Friedrich-Alexander-Universität
Erlangen-Nürnberg

Modulhandbuch

für den Studiengang

Bachelor of Science Informatik

(Prüfungsordnungsversion: 20092)

Inhaltsverzeichnis

Algorithmen und Datenstrukturen (93050) ... 10
Algorithmik kontinuierlicher Systeme (93000) ... 14
Berechenbarkeit und Formale Sprachen (93010) ... 16
Einführung in das Software Engineering (93097) ... 18
Grundlagen der Logik in der Informatik (93072) ... 20
Grundlagen der Rechnerarchitektur und -organisation (93080) ... 23
Grundlagen der Schaltungstechnik (93091) ... 25
Grundlagen der Technischen Informatik (93110) ... 27
Implementierung von Datenbanksystemen (93020) ... 29
Konzeptionelle Modellierung (93130) ... 32
Mathematik für INF 1 (67630) ... 34
Mathematik für INF 2 (67640) ... 36
Mathematik für INF 3 (67650) ... 38
Mathematik für INF 4 (67660) ... 40
Parallele und Funktionale Programmierung (93040) ... 42
Rechnerkommunikation (93150) ... 43
Systemprogrammierung (93180) ... 45
Theorie der Programmierung (93121) ... 47
Nebenfach
- Nebenfach Physik (1818) ... 50
- Nebenfach Soziologie (1821) ... 53

Nebenfach Astronomie
- Astronomie (66292) ... 56

Nebenfach Betriebswirtschaftslehre
- Business Process Management (83467) ... 59
- BWL für Ingenieure (82570) ... 61
- E-Business und E-Commerce (82397) ... 63
- Enterprise Content and Collaboration Management (86960) ... 65
- Marketing-Fallstudien (82491) ... 67
- Service Management und Service Engineering (82455) ... 69

Nebenfach Biologie
- Allgemeine Biologie I (62921) ... 71
- Allgemeine Biologie II (62922) ... 73
- Mikrobiologische Übungen (62982) ... 75
- Ökologie und Diversität B (62820) ... 77
- Zoologie (62940) ... 79

Nebenfach Chemie
- Allgemeine Organische Chemie (62963) ... 82
- Allgemeine und Anorganische Chemie (62060) ... 84

Nebenfach Englische Linguistik
- Basismodul II Linguistics (77011) ... 87
- Seminarmodul L-UF Linguistics (77230) ... 89

Nebenfach Geowissenschaften
- Rohstoffe und Nachhaltigkeit (64935) ... 92
- Z-Edu-Geo 1 - Das System Erde (64980) ... 94
- Z-Edu-Geo 2 - Der Mensch im geowissenschaftlichen Umfeld (64985) ... 96

Nebenfach Germanistische Linguistik
- Grundlagen der Germanistischen Linguistik 1 (Ling BM 1) (77301) ... 99
- Vertiefungsmodul Gegenwartssprache/DAF (Ling VM 2) (77432) ... 101
- Vertiefungsmodul Sprachwandel und Variation (Ling VM 1) (77402) ... 105

Nebenfach Mathematik
Algebra (65311) 108
Computational complexity (65092) 110
Discrete optimization I (65917) 112
Discrete optimization II (65933) 113
Diskretisierung und numerische Optimierung (65231) 115
Einführung in die Darstellungstheorie (65070) 117
Einführung in die Numerik (65210) 119
Elementare Zahlentheorie (65580) 121
Funktionalanalysis (65110) 122
Funktionentheorie I (65351) 124
Geometrie (65621) 125
Gewöhnliche Differentialgleichungen (65100) 126
Introduction to Statistics and Statistical Programming (48071) 128
Kryptographie I (65979) 130
Kryptographie II (65980) 132
Lineare und Kombinatorische Optimierung (65161) 133
Mathematische Modellierung Praxis (65255) 135
Mathematische Modellierung Theorie (65254) 136
Nichtlineare Optimierung (65150) 138
Numerical Aspects of Linear and Integer Programming (407487) 140
Numerik I für Ingenieure (64620) 141
Numerik II für Ingenieure (64631) 142
Operations Research 2 (65991) 143
Optimierung für Ingenieure (44050) 144
Optimierung für Ingenieure mit Praktikum (44060) 146
Optimization in industry and economy (65923) 149
Robuste Optimierung 1 (65175) 151
Robust optimization II (65918) 152
Topologie (65080) 153
Nebenfach Artificial Intelligence in Biomedical Engineering
A look inside the human body - gait analysis and simulation (96837) 155
Bewegungsanalyse und biomechanische Grenzgebiete (47666) 157
Computational Neurotechnology / Numerische Neurotechnologie (44200) 159
Inertial Sensor Fusion (92358) 161
Interfacing the Neuromuscular system: Applications for Human/Machine Interfaces and Neurophysiology (44157) 164
Legged Locomotion of Robots + Laborprojekt (LLR-L) (47657) 166
Legged Locomotion of Robots (LLR) (47656) 168
Research Project on Intelligent Sensorimotor Systems (47687) 170
Nebenfach Medizinische Informatik
Biometrie und Epidemiologie (22101) 173
Nebenfach Philosophie
Basismodul Philosophie (75330) 176
Einführung in die Philosophie (75290) 178
Grundkurs Praktische Philosophie (75310) 179
Nebenfach Sinologie
Modernes Chinesisch 1 (76003) 181
Modernes Chinesisch 2 (76012) 182
Nebenfach Ökonomie
Betriebliches Rechnungswesen I (74860) 184
Betriebswirtschaftslehre I (74810) 185
Betriebswirtschaftslehre II (74830) 186

Einführung in die Volkswirtschaftslehre (74820) 187
International Business Ethics I (38051) 188
Makroökonomie (82070) 191
Mikroökonomie (74840) 193
Monetäre Ökonomik (75060) 195
Sozialpolitik (74870) 196
Strategisches Management (74790) 197
Nebenfach EEI: Allgemeine Elektrotechnik
Werkstoffkunde für EEI (95610) 199
Nebenfach EEI: Automatisierungstechnik
Grundlagen der Elektrotechnik III (92580) 203
Regelungstechnik B (Zustandsraummethoden) (97060) 205
Nebenfach EEI: Elektrische Energie- und Antriebstechnik
Grundlagen der elektrischen Energieversorgung (92542) 209
Grundlagen der Elektrotechnik I (92560) 211
Grundlagen der Elektrotechnik II (92570) 213
Nebenfach EEI: Informationstechnik
Grundlagen der Nachrichtenübertragung (392436) 216
Signale und Systeme I (92681) 218
Signale und Systeme II (92682) 220
Nebenfach EEI: Mikroelektronik
Entwurf integrierter Schaltungen I (96590) 223
Schaltungstechnik (92660) 225
Nebenfach EEI: Leistungselektronik
Halbleiterbauelemente (92590) 228
Leistungselektronik (96630) 230
Regelungstechnik A (Grundlagen) (92650) 232
Nebenfach Maschinenbau: Grundlagen der Produktentwicklung
Grundlagen der Produktentwicklung (94711) 235
Technische Darstellungslehre 2 (94591) 241
Technische Produktgestaltung (97110) 245
Nebenfach Maschinenbau: Fertigungsautomatisierung und Produktionstechnik
Produktionstechnik I und II (94570) 251
Nebenfach Maschinenbau: Technische Mechanik
Dynamik starrer Körper (94500) 255
Statik und Festigkeitslehre (94660) 257
Nebenfach Maschinenbau: Qualitätsmanagement und Messtechnik
Grundlagen der Messtechnik (94510) 261
Schwerpunkt Physiologie
Anatomie und Physiologie für Nichtmediziner (22800) 271
Nebenfach Maschinenbau: Computerintegrierte Produktion
Automotive Engineering I (95340) 274
Die Werkzeugmaschine als mechatronisches System (95270) 276
Grundlagen der Robotik (94951) 278
Handhabungs- und Montagetechnik (97121) 280
Industrie 4.0 - Anwendungsszenarien in Design und Engineering (94947) 282
Industrie 4.0 - Anwendungsszenarien in Produktion und Service (94946) 284
Industrie 4.0 für Ingenieure (319238) 286
Integrated Production Systems (97123) 288
International Supply Chain Management (94920) 290
Machine Learning for Engineers II: Advanced Methods (95068) 292
Machine Learning for Engineers I - Introduction to Methods and Tools (95067) 294
Multimedia Security (330467) 296

Produktionssystematik (97101)..... 298
Robotics Frameworks (92880)..... 300
Technische Grundlagen des ressourcenschonenden und intelligenten Wohnens (94940)..... 302
Psychologie als Nebenfach für Informatik
Anwendungsmodul (78389)..... 305
Einführungsmodul (78386)..... 307
Grundlagenmodul (78388)..... 308
Praktikum Informatik
Applied Software Engineering Bachelor-Praktikum (93094)..... 312
Grafik-Praktikum Game Programming (240715)..... 314
Hackerpraktikum Bachelor (93192)..... 315
HPC Software Projekt (695344)..... 317
IoT Security (93199)..... 319
Mobile Application Development and Security (93203)..... 321
Nailing your Thesis (VUE+PROJ 10-ECTS) (917928)..... 323
NWERC Praktikum (93129)..... 325
The AMOS Project (SD Role, PROJ 10 ECTS) (93142)..... 329
Praktikum angewandte Systemsoftwaretechnik (113845)..... 331
Praktikum Enterprise Computing (594684)..... 332
Praktikum Entwicklung digitaler Lernstationen (93205)..... 333
Praktikum: Entwicklung interaktiver eingebetteter Systeme (716025)..... 334
Praktikum: Entwicklung interaktiver eingebetteter Systeme (93197)..... 337
Praktikum Künstliche Intelligenz (BSc) (93187)..... 340
Praktikum Lego Mindstorms (278855)..... 341
Praktikum Mustererkennung (93155)..... 343
Project Deep Learning in Multimedia Forensics (47635)..... 345
Rechnerarchitekturen für Deep-Learning Anwendungen (93111)..... 347
Secure Web Development (344655)..... 349
Softwareentwicklungspraktikum Lehramt (93162)..... 351
Supercomputing Praktikum (182798)..... 353
Hauptseminar
Advanced Competitive Programming (93655)..... 356
Automaten über unendlichen Wörtern (93126)..... 358
Blender Seminar (921878)..... 360
Geschichte der Rechentechnik (47637)..... 362
Green AI - AI for sustainability and sustainability of AI (47614)..... 364
IT-Sicherheits-Seminar (396551)..... 366
Konzepte von Betriebssystem-Komponenten (44590)..... 367
Iterative Lösungsverfahren für lineare und nichtlineare Gleichungssysteme (93096)..... 368
Maschinelles Lernen: Einführung (93127)..... 370
Nailing your Thesis (VUE 5-ECTS) (304439)..... 373
Neuartige Rechnerarchitekturen (941318)..... 375
The AMOS Project (PO Role, SEMI 5 ECTS) (93144)..... 377
Nailing your Thesis (PROJ 5-ECTS) (404439)..... 379
Seminar Applied Software Engineering (93186)..... 381
Recent Advances in Cryptography (93179)..... 382
Research Data Skills (93188)..... 383
Cryptography in Secure Messaging: Understanding and Enhancing Signal (93204)... 385
Seminar Algorithmische Schönheiten - Algorithms Unplugged (871263)..... 387
Seminar: Ausgewählte Kapitel aus "The Art of Computer Programming" (886484)..... 389
Seminar Einführung in die Kryptografie (153330)..... 391

Seminar Energieinformatik (93656)....392
Seminar 'Hallo Welt!' für Fortgeschrittene (319374).... 394
Seminar Koalgebraische Logik (95790).... 396
Seminar Kommunikationssysteme (B.Sc.) (609624)....399
Seminar Machine Learning and Data Analytics for Industry 4.0 (903776).... 401
Seminar Sprachtechnologie für Sprachpathologien (349413)....403
Seminar SystemC (716001).... 404
Seminar Theoretische Informatik (863761).... 406
Seminar Wissensrepräsentation und -verarbeitung (635405).... 408
Systems- and Networks-on-a-Chip für INF (834345)....410
Themen der Kategorientheorie (819238).... 412
Vertiefungsrichtung Datenbanksysteme
Datenbanken in Rechnernetzen und Transaktionssysteme (681735)....415
eBusiness Technologies und Evolutionäre Informationssysteme (710850).... 418
Knowledge Discovery in Databases mit Übung (43961)....422
Vertiefungsrichtung Diskrete Simulation
Simulation und Modellierung I (97090).... 426
Vertiefungsrichtung Elektronik und Informationstechnik
Globale Navigationssatellitensysteme (96401).... 430
Kommunikationsstrukturen (96801)....432
Satellitengestützte Ortsbestimmung (652213).... 434
Satellitenkommunikation (43460).... 436
Vertiefungsrichtung Hardware-Software-Co-Design
Cyber-Physical Systems (636348).... 441
Eingebettete Systeme (Vorlesung mit erweiterten Übungen) (773774).... 443
Eingebettete Systeme (Vorlesung mit Übungen) (604896)....446
Hardware-Software-Co-Design (502509).... 448
Hardware-Software-Co-Design (Vorlesung mit erweiterter Übung) (292952).... 451
Parallele Systeme (43510).... 454
Parallele Systeme (Vorlesung mit erweiterten Übungen) (740665).... 457
Reconfigurable Computing (43190).... 460
Reconfigurable Computing (Lecture with Exercises) (741941)....463
Reconfigurable Computing (Lecture with Extended Exercises) (110334).... 466
Verifikation digitaler Systeme (92280)....469
Vertiefungsrichtung Kommunikationssysteme
Kommunikationssysteme (43950).... 472
Modellierung, Optimierung und Simulation von Energiesystemen (858896)....474
Vertiefungsrichtung Künstliche Intelligenz
Künstliche Intelligenz I (894856)....477
Künstliche Intelligenz II (532733)....479
Logik-Basierte Sprachverarbeitung (675137)....481
Logik-basierte Wissensrepräsentation für mathematisch/technisches Wissen (221236)....483
Wissensrepräsentation und -verarbeitung (93134).... 485
Vertiefungsrichtung Mustererkennung
Biomedizinische Signalanalyse (23070)....488
Deep Learning for Beginners (93330)....492
Diagnostic Medical Image Processing (44150)....494
Human Computer Interaction (645618)....496
Interventional Medical Image Processing (44140).... 499
Introduction to Machine Learning (65718).... 501
Introduction to Network Science (93340)....504
Maschinelles Lernen für Zeitreihen (428256)....506

Music Processing Analysis - Lecture and Exercise (639119)..... 508
Pattern Recognition (44130)..... 511
Speech and Language Processing (44455).....514
Vertiefungsrichtung Programmiersysteme
Grundlagen des Übersetzerbaus (44240).....517
Mainframe Programmierung (505241)..... 525
Mainframe Programmierung II (93182)..... 527
Vertiefungsrichtung Rechnerarchitektur
Architekturen von Superrechnern (44460).....530
CPU Entwurf mit VHDL (CPU) (211243)..... 532
CPU Entwurf mit VHDL (Schwerpunkt VHDL) (436348).....534
Heterogene Rechnerarchitekturen Online (275245)..... 536
Rechnerarchitektur (798810).....539
Rechnerarchitektur (Vorlesung mit Übung und Rechnerübung) (333815)..... 541
Vertiefungsrichtung Software Engineering
Advanced Design and Programming (5-ECTS) (97008).....544
Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML) (510375).....546
Datenbank Praxis (93002).....548
Grundlagen des Software Engineering (93550).....550
Mainframe@Home (93183)..... 552
Nailing your Thesis (PROJ 5-ECTS) (580491)..... 554
Nailing your Thesis (VUE 5-ECTS) (480491)..... 556
The AMOS Project (PO Role, VUE 5 ECTS) (93145)..... 558
The AMOS Project (SD Role, VUE 10 ECTS) (93143).....560
Commercial Open Source Startups (OSS-COSS) (93184).....562
Praktische Softwaretechnik (57025).....564
Product Management (93198).....566
Software Projektmanagement (312443).....567
Testen von Softwaresystemen (189989).....569
Vertiefungsrichtung Systemsimulation
Advanced Programming Techniques (465562)..... 572
Programmiertechniken für Supercomputer (43740).....574
Programming Techniques for Supercomputers (Lecture and Tutorial) (278169).....576
Simulation und Wissenschaftliches Rechnen 1 (43370).....578
Vertiefungsrichtung Theoretische Informatik
Advanced Mechanized Reasoning in Coq (93169).....581
Algebra des Programmierens (861501)..... 582
Algebraische und Logische Aspekte der Automatentheorie (93079)..... 584
Approximationsalgorithmen (247639).....586
Beschreibungslogik und formale Ontologien (806144)..... 588
Effiziente kombinatorische Algorithmen (843472).....590
Ereignisgesteuerte Systeme (93540)..... 592
Formale Methoden der Softwareentwicklung (151316)..... 594
Kommunikation und Parallele Prozesse (173107).....596
Modallogik (984981)..... 598
Monad-Based Programming (845618)..... 600
Praktische Semantik von Programmiersprachen (599478).....603
Randomisierte Algorithmen (164985).....605
Swarm Intelligence (44500).....607
Vertiefungsrichtung Verteilte Systeme und Betriebssysteme
Betriebssysteme (Vorlesung mit erweiterten Übungen) (820947).....610
Betriebssysteme (Vorlesung mit Übungen) (150033)..... 613

Betriebssystemtechnik (V+EÜ) (350752)..... 616
Betriebssystemtechnik (Vorlesung mit Übungen) (406841)..... 618
Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme (939179)..... 620
Echtzeitsysteme mit erweiterten Übungen (179490)..... 623
Echtzeitsysteme (Vorlesung mit Übungen) (707303)..... 628
Middleware - Cloud Computing (Vorlesung mit erweiterten Übungen) (722831)..... 633
Middleware - Cloud Computing (Vorlesung mit Übungen) (557235)..... 636
Verlässliche Echtzeitsysteme (Vorlesung mit Übungen) (876012)..... 639
Verteilte Systeme (Vorlesung mit erweiterten Übungen) (743260)..... 642
Verteilte Systeme (Vorlesung mit Übungen) (649073)..... 645
Virtuelle Maschinen (Vorlesung mit Übungen) (202041)..... 648
Virtuelle Maschinen (Vorlesung mit Übung und Laborübung) (462793)..... 650
Web-basierte Systeme (93087)..... 652
Vertiefungsrichtung IT-Sicherheit
Angewandte IT-Sicherheit (326311)..... 655
Forensische Informatik (792501)..... 657
Human Factors in Security and Privacy (658644)..... 659
Security and Privacy in Pervasive Computing (327615)..... 662
Security in Embedded Hardware (172338)..... 664
Software Exploitation (93098)..... 667
Vertiefungsrichtung Informatik in der Bildung
Didaktik der Informatik I (93210)..... 669
Didaktik der Informatik II (93220)..... 672
Praktikum Informatik in der Bildung (PIB) (93149)..... 674
Vertiefungsrichtung Medizinische Informatik
Cognitive Neuroscience for AI Developers (44445)..... 676
Einführung in die Medizinische Informatik (93300)..... 678
Informationssysteme im Gesundheitswesen (22910)..... 681
Wissensbasierte Systeme in der Medizin 1 (22850)..... 683
Vertiefungsrichtung Kryptographie
Cryptocurrencies (93006)..... 686
Cryptographic Communication Protocols (93206)..... 688
Einführung in die moderne Kryptographie (93015)..... 690
Introduction to Privacy (93680)..... 692
Vertiefungsrichtung Visual Computing
Computational Visual Perception (93173)..... 694
Computer Graphics (43821)..... 696
Computergraphik (Vorlesung mit Übung und Praktikum) (43394)..... 699
Geometric Modeling (796399)..... 702
Globale Beleuchtungsberechnung (43395)..... 704
Grundlagen empirischer Forschungsmethoden in der medialen Interaktion (93176)... 706
Informationsvisualisierung (299892)..... 708
Interaktive Computergraphik (43391)..... 711
Neural Graphics and Inverse Rendering (93086)..... 713
Physically-based Simulation in Computer Graphics (43385)..... 714
Scientific Visualization (43722)..... 716
Visual Computing in Medicine (44481)..... 718
Visualization (93175)..... 722
Nebenfach Frankoromanistik für Informatikstudierende
Basismodul Einführung in die Französische Literaturwissenschaft (72203)..... 725
Basismodul Einführung in die Französische Sprachwissenschaft (72204)..... 727
Basismodul Französische Sprachpraxis 1 (72213)..... 729
Basismodul Französische Sprachpraxis 2 (72224)..... 731

Nebenfach Iberoromanistik für Informatikstudierende
Basismodul Spanische Literaturwissenschaft (72634)....734
Basismodul Spanische Sprachwissenschaft (72633)....736
Spanische Sprachpraxis 1 (72372)....738
Nebenfach Italoromanistik für Informatikstudierende
Basismodul Italienische Literaturwissenschaft (72624)....741
Basismodul Italienische Sprachpraxis 1 (72292)....743
Basismodul Italienische Sprachwissenschaft (72623)....745
Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik
Fertigungsmesstechnik I (97247)....748
Fertigungsmesstechnik II (96925)....758
Hauptseminar Messtechnik (607629)....762
Prozess- und Temperaturmesstechnik (97248)....764
Qualitätsmanagement II - Phasenübergreifendes Qualitätsmanagement (23030)....768
Qualitätsmanagement I - Qualitätstechniken für die Produktentstehung (95940)....770
Rechnergestützte Messtechnik (96930)....772

1	**Modulbezeichnung** 93050	**Algorithmen und Datenstrukturen** Algorithms and data structures	**10 ECTS**
2	Lehrveranstaltungen	Übung: R-D4 (2 SWS) Übung: T-A1 (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Dr.-Ing. Norbert Oster Patrick Sieber	

4	**Modulverantwortliche/r**	Prof. Dr. Michael Philippsen
5	**Inhalt**	• Grundlagen der Programmierung • Datenstrukturen • Objektorientierung • JAVA-Grundkenntnisse • Aufwandsabschätzungen • Grundlegende Algorithmen
6	**Lernziele und Kompetenzen**	<u>A - Fachkompetenz:</u> Die Studierenden... 1.) Grundlagen der Programmierung in Java • interpretieren Syntaxdiagramme für grundlegende Programmstrukturen und übertragen diese in entsprechenden Java-Code • deklarieren und verwenden Variablen mit adäquatem Java-Datentyp (primitive Typen, Reihungen, Zeichenketten) • überprüfen die Zulässigkeit der Variablendeklaration und -Wertzuweisung nach Java-Typ-Regeln • bestimmen den Datentyp und den Wert eines Java-Ausdrucks mit primitivem Datentyp und zugehörigen Operatoren • überführen einfache mathematische Ausdrücke in Java-Code • werten zusammengesetzte Bedingungen nach den Regeln der strikten bzw. faulen Auswertung für Java aus • konzipieren zu einer gegebenen Aufgabenstellung einen Algorithmus • implementieren einfache Algorithmen in Java unter Verwendung verschiedener Kontrollstrukturen • bestimmen die Gültigkeitsbereiche der Variablen anhand der Blockstruktur eines Java-Programms • strukturieren Java-Code in Methoden und entwickeln wiederverwendbare Funktionen 2.) Rekursion • beurteilen den Typ der Rekursion für gegebene Java-Methoden • entwerfen rekursive Algorithmen zu einer gegebenen Problemstellung unter Anwendung des Induktionsprinzips, des Teile-und-Herrsche-Prinzips sowie des Rücksetzverfahrens und implementieren diese jeweils in Java • entwickeln effizientere Lösungen, indem sie rekursive Methoden in endrekursive bzw. iterative Methoden umwandeln, implementieren diese jeweils in Java-Code und bewerten deren Laufzeit- und Speicherverbrauch

- bewerten und verbessern rekursive Lösungen unter Verwendung von Dynamischer Programmierung und implementieren diese in Java-Code

3.) Aufwandsanalyse

- analysieren den Laufzeitaufwand und den Speicherbedarf verschiedener Implementierungen
- klassifizieren den asymptotischen Laufzeitaufwand anhand der Komplexitätsklassen des O-Kalküls
- unterscheiden verschiedene Sortierverfahren (Blasensortierung, Sortieren durch Auswählen/Einfügen, Haldensortierung, Sortieren durch Verschmelzen/ Zerlegen/Fachverteilen) hinsichtlich ihres Laufzeit- und Speicherplatzbedarfs

4.) Objekt-Orientierte Programmierung in Java

- implementieren Java-Klassen gemäß textueller oder graphischer (UML) Spezifikation
- wenden Verfahren zur systematischen Ableitung von Klassen und Attributen (Hauptwortextraktion), ihren statischen Beziehungen (Vererbung, Polymorphie, Assoziationen) und ihrem dynamischen Zusammenspiel (CRC, Kollaboration) aus einer textuellen Problemstellung an und entwickeln so kleine objekt-orientierte Java-Programme
- instantiieren Klassen und verwenden Objektvariablen sachgerecht
- unterscheiden statische und dynamische Bindung gemäß Polymorphie-Konzept von Java und wenden die Erkenntnisse sachgerecht bei der Entwicklung eigener Applikationen an

5.) Robustes Programmieren

- wenden Checklisten an, um typische Programmierfehler im Vorfeld zu vermeiden oder nach der Programmierung zu identifizieren
- benutzen verschieden Möglichkeiten zur Absicherung gegen Fehlersituationen und zur Fehlerrückmeldung (Rückgabewert, Ausnahmebehandlung)
- wenden Junit zum Testen von Java-Programmen an
- setzen Verfahren und Werkzeuge zur systematischen Lokalisierung und Behebung von Programmfehlern an (Debugging) und verbessern ihre Lösungen auf diese Weise iterativ

6.) Elementare Datentypen

- übertragen eine Spezifikation in Form eines Abstrakten Datentyps (ADT) in ein gleichwertiges Java-Modul
- erstellen eine formale Spezifikation eines Datentyps in Form eines Abstrakten Datentyps (ADT) aus einer gegebenen textuellen Beschreibung
- verstehen die grundlegende Behälterdatentypen (Liste, Stapel, Schlange, Streutabelle) und deren Eigenschaften (insbesondere Laufzeit- und Speicherplatzbedarf ihrer Operationen)

		• verwenden generische Behälterdatentypen sachgerecht in eigenen Programmen • kennen die Verfügbarkeit generischer Behälterdatentypen in der Java-API und erschließen sich bei Bedarf selbst neue Datentypen sowie deren Funktionen aus der zugehörigen API-Spezifikation für die Verwendung in eigenen Programmen 7.) Bäume und Graphen • bewerten verschiedene Baum- und Graphdarstellungen hinsichtlich Zeitaufwand und Speicherbedarf • unterscheiden und klassifizieren die grundlegenden Baum-Arten (Suchbaum, AVL-Baum, Halde) • wenden die Grundoperationen (Einfügen, Suchen, Löschen, ggf. Restrukturieren) anhand von Beispieldaten auf gegebene Bäume artgerecht an • implementieren und verwenden verschiedene Baumstrukturen sachgerecht in eigenen Java-Programmen • führen verschiedene Durchlaufmöglichkeiten (Tiefensuche (DFS), Breitensuche (BFS)) für Graphen und Bäume auf Beispieldaten aus und setzen diese zielführend in eigenen Java-Programmen ein • wenden grundlegende Graphalgorithmen (Dijkstra, Floyd, Prim, Kruskal) auf Beispieldaten an und implementieren diese Verfahren in Java-Code <u>B - Selbst- und Sozialkompetenz:</u> Die Studierenden... • organisieren sich selbständig zu Gruppen und koordinieren in gegenseitiger Absprache den organisatorischen und technischen Ablauf der Gruppenarbeiten • kommunizieren und erarbeiten gemeinsam Lösungen für theoretische Fragestellungen und praktische Programmieraufgaben in Rahmen von Gruppenaufgaben • planen und wenden zielgerichtet Maßnahmen zu gegenseitigen Qualitätssicherung der eingereichten Lösungen an (prüfen wechselseitig die Gruppenabgaben) • verantworten gemeinsam das Ergebnis ihrer Gruppenarbeit, deren Bewertung für beide Gruppenpartner gleichermaßen gilt
7	**Voraussetzungen für die Teilnahme**	Wichtiger Hinweis: Die Vorlesung fand letztmals im Wintersemester 2021/22 statt. Übungsbetrieb und Klausur werden vorerst noch in jedem Semester angeboten, allerdings endet das Angebot in naher Zukunft.
8	**Einpassung in Studienverlaufsplan**	Semester: 1
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20092
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten) Übungsleistung

11	**Berechnung der Modulnote**	Klausur (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 120 h Eigenstudium: 180 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Lehrbuch: Saake, Sattler: Algorithmen und Datenstrukturen - Eine Einführung mit JAVA

1	**Modulbezeichnung** 93000	**Algorithmik kontinuierlicher Systeme** Algorithms for continuous systems	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Ulrich Rüde
5	**Inhalt**	• Grundlagen kont. Datenstrukturen (Gleitpunktzahlen, Rundungsfehleranalyse und Kondition, Diskretisierung und Quantisierung, Abtasttheorem, FFT) • Algorithmische Lineare Algebra (direkte und iterative Verfahren für lin. Gleichungssysteme, Ausgleichsprobleme) • Datenstrukturen für geometrische Objekte, Interpolation, Approximation, Grundlagen geometrischer Modellierung, Volumen- und Flächenberechnung. • Kontinuierliche und diskrete Optimierung, nichtlineare Probleme. • Grundlagen der Simulation: Algorithmen zur Lösung von Differentialgleichungen
6	**Lernziele und Kompetenzen**	Die Studierenden erwerben fundierte Kenntnisse über Datenstrukturen und Algorithmen zur Behandlung kontinuierlicher Probleme. Die erworbenen Kompetenzen sind sowohl theoretisch-analytischer Art (Analyse von Komplexität, Konvergenz, Fehlerentwicklung) als auch von praktischer Natur (Implementierung der Algorithmen in einer objekt-orientierten Programmiersprache). Die Studierenden planen und bearbeiten kleine Programmierprojekte so, dass sie zeitgerecht fertig gestellt werden. Sie erwerben damit insbesondere die Grundlagen, die für ein vertieftes Studium in den Bereichen Systemsimulation, Mustererkennung, Graphischer Datenverarbeitung unabdingbar sind. Fachkompetenz Wissen Die Studierenden • geben die Definition von Gleitpunktzahlen wieder • reproduzieren Formeln zur Berechnung von Flächen und Volumina Verstehen Die Studierenden • erklären die Kondition von Problemen • veranschaulichen Methoden der Freiformflächenmodellierung • erläutern das Abtasttheorem und die Fouriertransformation Anwenden Die Studierenden • implementieren Algorithmen zur Lösung von linearen Gleichungssystemen • lösen Interpolations- und Approximationsaufgaben • berechnen iterativ Lösungen von nichtlinearen Gleichungen Analysieren

		Die Studierenden • klassifizieren Optimierungsprobleme • erforschen lineare Ausgleichsprobleme Lern- bzw. Methodenkompetenz Die Studierenden • ° lösen Aufgaben der Algorithmen kontinuierlicher Probleme in Gruppenarbeit
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) Übungsleistung
11	**Berechnung der Modulnote**	Klausur (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93010	**Berechenbarkeit und Formale Sprachen** Theory of computation and formal languages	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Berechenbarkeit und Formale Sprachen (4 SWS) Übung: UE-BFS (2 SWS)	5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr. Rolf Wanka Benjamin Seiler Martin Meyer ter Vehn Johanna Zaviska Dominik Pysch Matthias Kergaßner Sebastian Dietlmeier	

4	**Modulverantwortliche/r**	Prof. Dr. Rolf Wanka
5	**Inhalt**	• Registermaschinen und Turingmaschinen als Modelle des Berechenbaren, die Church-Turing-These und unentscheidbare Probleme • NP-Vollständigkeit und das P-NP-Problem • Endliche Automaten • Grammatiken und die Chomsky-Hierarchie • Kontextfreie Grammatiken und Kontextfreie Sprachen • Kellerautomaten
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben fundierte Kenntnisse über die Grenzen der Berechenbaren, insbesondere lernen sie, wie man beweist, dass bestimmte Aufgaben unlösbar sind bzw. dass sie vermutlich nicht schnell gelöst werden können, und wenden diese Kenntnisse an; • lernen die wesentlichen Techniken kennen, mit denen man Programmiersprachen beschreiben und syntaktisch korrekte Programme erkennen kann, und wenden diese auf Beispiele an; • erwerben fundierte Kenntnisse in den Beweis- und Analyse-Methoden der algorithmisch orientierten Theoretischen Informatik und wenden diese an.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 3
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) Übungsleistung
11	**Berechnung der Modulnote**	Klausur (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	nur im Wintersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• I. Wegener. Theoretische Informatik. • J. Hopcroft, J. Ullman. Introduction to Automata Theory, Languages and Computation. • U. Schöning. Theoretische Informatik - kurz gefasst.

1	**Modulbezeichnung** 93097	**Einführung in das Software Engineering** Introduction to software engineering	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Introduction to Software Engineering (SWS) Übung: Introduction to Software Engineering Exercises (SWS)	- -
3	Lehrende	Prof. Dr.-Ing. Andreas Maier	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Andreas Maier
5	**Inhalt**	• Einführung in die einzelnen Phasen der Softwareentwicklung: Anforderungsanalyse, Spezifikation, Entwurf, Implementierung, Test, WartungProzessmodelle • Prozessmodelle • Agile Softwareentwicklung • Anforderungsanalyse und -verwaltung • Modellierung von Systemen (u.a. mit UML) • Software-Architekturen und Designmuster • Teststrategien • Umgang mit Software-Alterung • Projektmanagement • Software-Engineering im Bereich Machine Learning • Refactoring zur Unterstützung der Wartungsphase
6	**Lernziele und Kompetenzen**	Die Studierenden • Beschreiben Prozessmodelle und unterscheiden plangesteuerte (wie das Wasserfall- und V-Modell) und agile Prozessmodelle (wie XP, Scrum, RUP und Kanban) • Erläutern verschiedene Techniken der Anforderungsanalyse und –Ermittlung (wie Endliche Zustandsautomaten, Petri-Netze, Use Cases, User Stories) und wenden diese für plan-gesteuerte und agile Prozesse an • Stellen die Unterschiede zwischen agilem und plan-gesteuertem Requirements-Engineering dar • Verstehen und erläutern UML-Diagramme (wie Use Case-, Klassen-, Sequenz- und Kommunikationsdiagramme) und wenden diese auf praktische Beispiele der Objektorientierung an • Reproduzieren allgemeine Entwurfslösungen wiederkehrender Probleme des Software-Engineerings und wenden diese an • Wenden funktionale und strukturelle Testansätze an • Erklären Methoden zur Änderung und Weiterentwicklung von Software • Beschreiben Ansätze für das Projekt-Management von Softwareprojekten • Erläutern wie Methoden des Maschinellen Lernens für Software-Engineering eingesetzt werden können

7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 3
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	
16	**Literaturhinweise**	• Software Engineering, Ian Sommerville, 10. Auflage, 2016 • Software-Engineering Kompakt, Anja Metzner, 2020 • Handbook of Software Engineering, Sungdeok Cha, Richard N. Taylor, Kyochul Kang (Hrsg.), 2019

1	**Modulbezeichnung** 93072	**Grundlagen der Logik in der Informatik** Foundations of logic in informatics	**5 ECTS**
2	Lehrveranstaltungen	Übung: Intensivübung zu Grundlagen der Logik in der Informatik (2 SWS) Übung: Übungen zu Grundlagen der Logik in der Informatik (2 SWS) Vorlesung: Grundlagen der Logik in der Informatik (2 SWS)	- - 5 ECTS
3	Lehrende	Thorsten Wißmann Max Ole Elliger Zisis Erkelentzis David Wegmann Leonard Baumgartner Silas Kuder Wilhelm Steinleitner Leon Vatthauer Prof. Dr. Lutz Schröder	

4	**Modulverantwortliche/r**	Prof. Dr. Lutz Schröder
5	**Inhalt**	Aussagenlogik: • Syntax und Semantik • Automatisches Schließen: Resolution • Formale Deduktion: Korrektheit, Vollständigkeit Prädikatenlogik erster Stufe: • Syntax und Semantik • Automatisches Schließen: Unifikation, Resolution • Quantorenelimination • Anwendung automatischer Beweiser • Formale Deduktion: Korrektheit, Vollständigkeit
6	**Lernziele und Kompetenzen**	• Erwerb fundierter Kenntnisse zu den Grundlagen und der praktischen Relevanz der Logik mit besonderer Berücksichtigung der Informatik; • Verstehen und Erklären des logischen Schließens; • Einübung in das logische und wissenschaftliche Argumentieren, Aufstellen von Behauptungen und Begründungen; • Kritische Reflexion von Logikkalkülen, insbesondere hinsichtlich Entscheidbarkeit, Komplexität, Korrektheit und Vollständigkeit; • Erstellung und Beurteilung von Problemspezi;kationen (Kohärenz, Widerspruchsfreiheit) und ihre Umsetzung in Logikprogramme; • Beherrschung der praktischen Aspekte der Logikprogrammierung. Fachkompetenz Wissen Die Studierenden

		geben Definitionen zur Syntax und Semantik der verwendeten Logiken wieder beschreiben grundlegende Deduktionsalgorithmen geben Regeln der verwendeten formalen Deduktionssysteme wieder Verstehen Die Studierenden erläutern das Verhältnis zwischen Syntax, Semantik und Beweistheorie der verwendeten Logiken erklären die Funktionsprinzipien grundlegender Deduktionsalgorithmen erläutern die Funktionsweise automatischer Beweiser erläutern grundlegende Resultate der Metatheorie der verwendeten Logiken und deren Bedeutung Anwenden Die Studierenden wenden Deduktionsalgorithmen auf konkrete Deduktionsprobleme an formalisieren Anwendungsprobleme in logischer Form und verwenden automatische Beweiser zur Erledigung entstehender Beweisziele führen einfache formale Beweise manuell Analysieren Die Studierenden führen einfache metatheoretische Beweise, inbesondere durch syntaktische Induktion Lern- bzw. Methodenkompetenz Die Studierenden beherrschen das grundsätzliche Konzept des Beweises als hauptsächliche Methode des Erkenntnisgewinns in der theoretischen Informatik. Sie überblicken abstrakte Begriffsarchitekturen. Sozialkompetenz Die Studierenden lösen abstrakte Probleme in Gruppenarbeit.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 1;
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Schöning, U.: Logik für Informatiker.

		Heidelberg: Spektrum Akademischer Verlag, 2000 Barwise, J., and Etchemendy, J.: Language, Proof and Logic; CSLI, 2000. Huth, M., and Ryan, M.: Logic in Computer Science; Cambridge University Press, 2000.

1	**Modulbezeichnung** 93080	**Grundlagen der Rechnerarchitektur und -organisation** Foundations of computer architecture and computer organisation	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Dietmar Fey
5	**Inhalt**	Ziel der Vorlesung ist, die Grundlagen beim Aufbau eines Rechners zu vermitteln. Dies beinhaltet die Grundkomponenten, wie das Leitwerk, das Rechenwerk, das Speicherwerk und das Ein-/Ausgabewerk. Ausgehend vom klassischen von Neumann-Rechner wird der Bogen bis zu den Architekturen moderner Rechner und Prozessoren geschlagen. Grundprinzipien der Ablaufstuerung bei der Berarbeitung von Befehlen werden ebenso behandelt wie Aufbau und Funktionsweise eines Caches und die Architektur von Speichern im Allgemeinen. Das Konzept der Mikroprogrammierung wird erläutert. Ferner wird der Einstieg in die hardwarenahe Programmierung moderner CPUs mittels Assembler vorgestellt und erprobt. Aufbau und Funktionsweise peripherer Einheiten und Bussysteme werden ebenfalls behandelt. Die Studierenden sollen am Ende der Vorlesung den Aufbau und die Funktionsweise der Architektur eines Rechners, z.B. eines PCs, und des darin enthaltenen Prozessors nicht nur kennen, sondern auch die Gründe für deren Zustandekommen verstanden haben.
6	**Lernziele und Kompetenzen**	Nach dem Besuch der Lehrveranstaltung kennen die Studierenden die Grundkomponenten eines Rechners, z. B. eines PCs, und können diese auch im Zusammenspiel als Gesamtsystem erklären, sowie die Eigenheiten verschiedener Architekturen diskutieren. Sie können die Funktionsweise von Grundkomponenten wie Leitwerk, Rechenwerk, Speicherwerk, Ein-/Ausgabewerk, Bussystemen, sowie peripherer Komponenten erläutern und in die Struktur eines Computerssystems einordnen. Sie kennen den Aufbau von Caches, bzw. von Speichern im Allgemeinen und verstehen die Funktionsweise der Ablaufstuerung, insbesondere in Bezug auf die Abarbeitung von Befehlen. Weiterhin können die Studierenden Konzepte der Mikroprogrammierung unterscheiden, sowie hardwarenahe Programme in Assembler verstehen, modifizieren und erstellen.
7	**Voraussetzungen für die Teilnahme**	Keine

8	**Einpassung in Studienverlaufsplan**	Semester: 2
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Hennessy/Patterson: Computer Architecture - A quantitative approach, 4.Auflage, 2006, MorganKaufmann. Patterson/Hennessy: Computer Organization & Design, 4.Auflage, 2008, MorganKaufmann. Stallings, Computer Organization & Architecture, 8.Auflage, 2009, Prentice Hall. Märtin, Rechnerarchitekturen, 2001, Fachbuchverlag Leipzig.

1	**Modulbezeichnung** 93091	**Grundlagen der Schaltungstechnik** Foundations of circuit technology	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Albert Heuberger
5	**Inhalt**	Im Modul werden grundlegende Zusammenhänge elektrotechnischer Systeme, welche die Grundlage fast aller digitalen Datenverarbeitungssystemen bilden, behandelt. Zu Beginn werden elektrotechnischen Grundbegriffe und mathematische Grundlagen vermittelt. Daraufhin werden die Techniken zur Analyse von Gleich- und Wechselstromnetzwerken erläutert. Die grundlegenden elektrotechnischen Bauelemente Widerstand, Kondensator und Spule werden eingeführt und ihre Eigenschaften untersucht. Nachfolgend werden nichtlineare Bauelemente, wie Diode, Transistor und Operationsverstärker betrachtet und analysiert. Die Netzwerkanalyse wird anschließend auf Schaltvorgänge ausgeweitet. Außerdem wird das Funktionsprinzip von CMOS-Schaltungen erläutert und einfache digitale logische Grundschaltungen behandelt. Abschließend wird ein Überblick über Prinzipien der Datenspeicherung auf mikroelektronischer Basis gegeben, sowie Schaltungen zu Digital-Analog- und Analog-Digital-Wandlung vorgestellt und diskutiert. Elektrotechnische Grundlagen • Mathematische Grundlagen • Netzwerkanalyse Gleichstromfall • Netzwerkanalyse Wechselstromfall Elektronische Bauelemente • Widerstand, Kondensator, Spule (passiv) • Diode, Transistor, Operationsverstärker (aktiv) Einfache dynamische Vorgänge in Schaltungen (Schaltvorgänge) Digitale Schaltungstechnik • Digital-Analog-Wandler • Analog-Digital-Wandler Schaltungen zur Realisierung der Schnittstelle zwischen Analog- und Digitaltechnik • Funktionsprinzip von CMOS-Schaltungen • Einführung logischer Grundschaltungen in CMOS • Prinzipien mikroelektronischer Datenspeicher
6	**Lernziele und Kompetenzen**	Die Studierenden: • lernen elektrotechnische Zusammenhänge kennen • verstehen die Grundprinzipien elektrotechnischer Netzwerke • verstehen die Funktionsweise der elektrischen Basisbausteine für die Informatik • verstehen die Berechnung einfacher Schaltungen im Gleichstromfall, Wechselstromfall und bei Schaltvorgängen

		• erklären die Funktionsweise einfacher digitaler Schaltungen • entwickeln ein Verständis für die Bedeutung grundlegender elektrotechnischer Zusammenhänge für die Informatik • entwickeln ein methodisches Vorgehen zur Problemlösung in der Schaltungsanalyse
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 2
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20092
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) • Lehrveranstaltung findet nicht mehr statt • Studon-Kurs mit Material verfügbar (für Zutritt bitte eine Mail an like-info@fau.de) • Prüfung ist bis SoSe 2024 möglich
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Tietze, U. ; Schenk, Ch. : Halbleiter-Schaltungstechnik. Berlin: Springer.

1	**Modulbezeichnung** 93110	**Grundlagen der Technischen Informatik** Foundations of computer engineering	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Grundlagen der Technischen Informatik (4 SWS) Übung: Praktische Übungen zu Grundlagen der Technischen Informatik - Gruppe 5 (Mittwoch 9:00 Uhr) (0 SWS) Übung: Übungen zu Grundlagen der Technischen Informatik (2 SWS)	5 ECTS - 2,5 ECTS
3	Lehrende	Prof. Dr.-Ing. Jürgen Teich Dr.-Ing. Stefan Wildermann Mark Deutel Marcel Brand Jan Spieck Christian Heidorn	

4	**Modulverantwortliche/r**	Joachim Falk Prof. Dr.-Ing. Jürgen Teich
5	**Inhalt**	Aufbau und Prinzip von Rechnern, Daten und ihre Codierung, Boolesche Algebra und Schaltalgebra, Schaltnetze (Symbole, Darstellung), Optimierung von Schaltnetzen (Minimierung Boolescher Funktionen), Realisierungsformen von Schaltnetzen (ROM, PLA, FPGA), Automaten und Schaltwerke (Moore/Mealy, Zustandscodierung und -minimierung), Flipflops, Register, Zähler, Speicher (RAM, ROM), Taktung und Synchronisation, Realisierungsformen von Schaltwerken, Realisierung der Grundrechenarten Addition/Subtraktion, Multiplikation und Division, Gleitkommazahlen (Darstellung, Fehler, Rundung, Standards, Einheiten), Steuerwerksentwurf, Spezialeinheiten und Co-Prozessoren, Mikrocontroller; vorlesungsbegleitende Einführung und Beschreibung der Schaltungen mit VHDL.
6	**Lernziele und Kompetenzen**	Fachkompetenz - Wissen • Die Studierenden veranschaulichen fundierte theoretische und praxisorientierte Grundlagen der Informationstheorie, Rechnerarithmetik, Digitaltechnik und des Schaltungsentwurfs. • Die Studierenden führen den Entwurf, die Synthese und das Testen von digitalen Schaltungen auf programmierbarer Hardware (FPGAs) durch. Fachkompetenz - Verstehen • Die Studierenden verstehen, dass Hardware heutzutage mit Software am Rechner entwickelt und simuliert wird. • Die Studierenden verstehen den Schaltungsentwurf mittels einer Beschreibungssprache (VHDL). Fachkompetenz - Anwenden • Die Studierenden erarbeiten und diskutieren verschiedene Lösungswege für die Datencodierung sowie den Entwurf und die Optimierung von digitalen Hardwareschaltungen. Selbstkompetenz

		• Die Studierenden erlernen die Fähigkeit, digitale Schaltungen und Systeme eigenständig zu konzipieren und zu implementieren.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 1
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten) Übungsleistung
11	**Berechnung der Modulnote**	Klausur (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/grundlagen-der-technischen-informatik-im-wintersemester

1	**Modulbezeichnung** 93020	**Implementierung von Datenbanksystemen** Implementation of database systems	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Implementierung von Datenbanksystemen (2 SWS) Übung: Implementierung von Datenbanksystemen -- Übung 3 (2 SWS) Übung: Implementierung von Datenbanksystemen -- Intensivierungsübung (2 SWS)	2,5 ECTS 2,5 ECTS -
3	Lehrende	Demian Vöhringer Felix Hanika Konstantin Müller	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Richard Lenz
5	**Inhalt**	Die Vorlesung führt ein in den Aufbau und die Architektur von Datenbanksystemen, die Modularisierung und Schichtenbildung mit Abstraktionen verwenden. Schwerpunkt sind deshalb systemtechnische Aspekte von Datenbanksystemen. Die Übungen vertiefen verschiedene Aspekte an Beispielrechnungen und erweitern gelegentlich auch noch den Stoff um einige Facetten (z.b. Mehrattribut-Zugriffspfade). Ausgangspunkt einer Reihe von aufeinander aufbauenden Abstraktionen ist die Speicherung von Daten auf Hintergrundspeichern. Die erste Abstraktion ist die Datei. Dann werden Sätze eingeführt und auf verschiedene Weisen in Blöcken organisiert (sequenziell, mit Direktzugriff, indexsequentiell). Das schließt die Organisation eines Blockpuffers und Zugriffspfade (Indexstrukturen) unterschiedlichen Typs ein. Als zweite große Abstraktion werden Datenmodelle eingeführt und hier insbesondere das relationale. Das ist bereits aus dem Modul "Konzeptionelle Modellierung" bekannt, wird hier aber aus einer ganz anderen Perspektive heraus entwickelt. Der zweite Teil befasst sich mit der Realisierung der Leistungen eines Datenbanksystems unter Verwendung der vorher eingeführten Sätze und Zugriffspfade ("top-down"). Das umfasst die Anfrageverarbeitung und -optimierung, aber auch die Mechanismen zur Protokollierung von Aktionen und zur Wiederherstellung von Datenbankzuständen nach einem Fehler oder Ausfall. Ein laufend vervollständigtes Schichtenmodell fasst abschließend die Aufgaben in einer Architektur für Datenbank-Verwaltungssysteme zusammen. Ziel des Moduls ist es also, ein grundlegendes Verständnis für den Aufbau und die Funktionsweise eines Datenbanksystems zu vermitteln.
6	**Lernziele und Kompetenzen**	Die Studierenden • kennen das Schichtenmodell eines Datenbankverwaltungssystems; • verstehen das Prinzip der Datenunabhängigkeit (Datenabstraktion); • beherrschen das Aufbauprinzip einer Software-Schicht; • unterscheiden die Begriffe "Datenbank", "Datenbanksystem" und "Datenbankverwaltungssystem"; • unterscheiden die Begriffe "Datenmodell" und "Schema";

		• zeigen das Konzept der blockorientierten Datei mit ihren Zugriffsoperationen auf; • unterscheiden einen Satz von einem Block; • erklären das Konzept der sequentiellen Satzdatei; • schildern das Prinzip der Wechselpuffertechnik; • charakterisieren den Schlüsselzugriff auf Sätze; • stellen Gestreute Speicherung (Hashing) auf der Basis von Blöcken (Buckets) dar; • formulieren die Funktionsweise des Virtuellen Hashings; • fassen die Funktionsweise eines B-Baums zusammen; • unterscheiden die Dienste eines B-Baums von denen des Hashings; • können für eine Folge von Schlüsselwerten einen B-Baum aufbauen; • unterscheiden einen B-Baum von einem B-Stern-Baum (B+-Baum); • veranschaulichen einen Bitmap-Index; • unterscheiden die Primär- und Sekundärorganisation von Sätzen; • zählen Ersetzungsstrategien der Pufferverwaltung auf und vergleichen sie; • benennen die Dienste einer Pufferverwaltung; • erklären die Konzepte "Seite" und "Segment" im Gegensatz zu "Block" und "Datei"; • unterscheiden direkte und indirekte Seitenzuordnung; • interpretieren in Programmiersprachen eingebettete Anfragesprachen und Datenbank-Unterprogrammaufrufe; • charakterisieren Datenbank-Transaktionen; • kennen die Aufrufe zur Definition von Transaktionen; • erläutern die spaltenweise Abspeicherung von Relationen; • diskutieren die algebraische Optimierung von Anfragen; • stellen Planoperatoren eines Datenbanksystems dar; • unterscheiden Planoperatoren für den Verbund; • beschreiben Kostenformeln für die Abschätzung von Anfrageausführungen; • schildern die verschiedenen Anomalien im Mehrbenutzerbetrieb; • beschreiben die Serialisierbarkeit von Transaktionen; • erläutern das Konzept der Sperren in Datenbanksystemen; • unterscheiden physische und logische Konsistenz; • kennen die vier Recovery-Klassen; • erläutern die verschiedenen Arten von Sicherungspunkten.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20092

10	**Studien- und Prüfungsleistungen**	Klausur mit MultipleChoice (90 Minuten)
11	**Berechnung der Modulnote**	Klausur mit MultipleChoice (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	KEMPER, Alfons ; EICKLER, André: Datenbanksysteme : Eine Einführung. 9., aktual. u. erweit. Aufl. München : Oldenbourg, 2013. ISBN 978-3-486-72139-3. Kapitel 7 bis 11 KEMPER, Alfons ; WIMMER, Martin: Übungsbuch Datenbanksysteme. 2., aktual. u. erweit. Aufl. München : Oldenbourg, 2009. ISBN 978-3-486-59001-2. Kapitel 7 bis 11 HEUER, Andreas ; SAAKE, Gunter: Datenbanken : Konzepte und Sprachen. 3., aktual. u. erw. Aufl. Bonn : mitp, 2007. - ISBN 3-8266-1664-2 HÄRDER, Theo ; RAHM, Erhard: Datenbanksysteme : Konzepte und Techniken der Implementierung. Berlin : Springer, 1999 - ISBN 3-540-65040-7 SAAKE, Gunter ; HEUER, Andreas: Datenbanken : Implementierungstechniken. 2., aktual. u. erw. Aufl. Bonn : mitp, 2005. ISBN 3-8266-1438-0

1	**Modulbezeichnung** 93130	**Konzeptionelle Modellierung** Conceptual modelling	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übungen zu Konzeptionelle Modellierung (WS23-06) (2 SWS) Vorlesung: Konzeptionelle Modellierung (2 SWS) Vorlesung: Introduction to Software Engineering (SWS) Übung: Introduction to Software Engineering Exercises (SWS)	2,5 ECTS 2,5 ECTS - -
3	Lehrende	Constantin Winkler David Haller Dominik Probst Alexander Seifert Tobias Bittner Prof. Dr.-Ing. Richard Lenz Prof. Dr.-Ing. Andreas Maier	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Richard Lenz
5	**Inhalt**	• Grundlagen der Modellierung • Datenmodellierung am Beispiel Entity-Relationship-Modell • Modellierung objektorientierter Systeme am Beispiel UML • Relationale Datenmodellierung und Anfragemöglichkeiten • Grundlagen der Metamodellierung • XML • Multidimensionale Datenmodellierung • Domänenmodellierung und Ontologien
6	**Lernziele und Kompetenzen**	Die Studierenden: • definieren grundlegende Begriffe aus der Datenbankfachliteratur • erklären die Vorteile von Datenbanksystemen • erklären die verschiedenen Phasen des Datenbankentwurfs • benutzen das Entity-Relationship Modell und das erweiterte Entity-Relationship Modell zur semantischen Datenmodellierung • unterscheiden verschiedene Notationen für ER-Diagramme • erläutern die grundlegenden Konzepte des relationalen Datenmodells • bilden ein gegebenes EER-Diagramm auf ein relationales Datenbankschema ab • erklären die Normalformen 1NF, 2NF, 3NF, BCNF und 4NF • definieren die Operationen der Relationenalgebra • erstellen Datenbanktabellen mit Hilfe von SQL • lösen Aufgaben zur Datenselektion und Datenmanipulation mit Hilfe von SQL • erklären die grundlegenden Konzepte der XML • erstellen DTDs für XML-Dokumente

		• benutzen XPATH zur Formulierung von Anfragen an XML-Dokumente • definieren die grundlegenden Strukturelemente und Operatoren des multidimensionalen Datenmodells • erklären Star- und Snowflake-Schema • benutzen einfache UML Use-Case Diagramme • benutzen einfache UML-Aktivitätsdiagramme • erstellen UML-Sequenzdiagramme • erstellen einfache UML-Klassendiagramme • erklären den Begriff Meta-Modellierung • definieren den Begriff der Ontologie in der Informatik • definieren die Begriffe RDF und OWL
7	**Voraussetzungen für die Teilnahme**	Gewünscht "Algorithmen und Datenstrukturen" und "Grundlagen der Logik und Logikprogrammierung"
8	**Einpassung in Studienverlaufsplan**	Semester: 1
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20092
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Elmasri, Ramez, and Sham Navathe. Grundlagen von Datenbanksystemen. Pearson Deutschland GmbH, 2009. - ISBN-10: 9783868940121 • Alfons Kemper, Andre Eickler: Datenbanksysteme : Eine Einführung. 6., aktualis. u. erw. Aufl. Oldenbourg, März 2006. - ISBN-10: 3486576909 • Bernd Oestereich: Analyse und Design mit UML 2.1. 8. Aufl. Oldenbourg, Januar 2006. - ISBN-10: 3486579266 • Ian Sommerville: Software Engineering. 8., aktualis. Aufl. Pearson Studium, Mai 2007. - ISBN-10: 3827372577 • Horst A. Neumann: Objektorientierte Softwareentwicklung mit der Unified Modeling Language. (UML). Hanser Fachbuch, März 2002. - ISBN-10: 3446188797 • Rainer Eckstein, Silke Eckstein: XML und Datenmodellierung. Dpunkt Verlag, November 2003. - ISBN-10: 3898642224

1	**Modulbezeichnung** 67630	**Mathematik für INF 1** Mathematics for INF 1	**7,5 ECTS**
2	Lehrveranstaltungen	Übung: IngMathC1U (2 SWS) Vorlesung: Mathematik für Ingenieure C1: INF, IP, ILS (4 SWS)	- 7,5 ECTS
3	Lehrende	PD Dr. Serge Kräutle	

4	**Modulverantwortliche/r**	apl.Prof.Dr. Martin Gugat
5	**Inhalt**	*Grundlagen* Aussagenlogik, Mengen, Relationen, Abbildungen *Zahlensysteme* natürliche, ganze, rationale und reelle Zahlen, komplexe Zahlen *Vektorräume* Grundlagen, Lineare Abhängigkeit, Spann, Basis, Dimension, euklidische Vektor- und Untervektorräume, affine Räume *Matrizen, Lineare Abbildungen, Lineare Gleichungssysteme* Matrixalgebra, Lösungsstruktur linearer Gleichungssysteme, Gauß-Algorithmus, inverse Matrizen, Matrixtypen, lineare Abbildungen, Determinanten, Kern und Bild, Eigenwerte und Eigenvektoren, Basis, Ausgleichsrechnung *Grundlagen Analysis einer Veränderlichen* Grenzwert, Stetigkeit, elementare Funktionen, Umkehrfunktionen
6	**Lernziele und Kompetenzen**	Die Studierenden • erklären grundlegende Begriffe und Strukturen der Mathematik • erklären den Aufbau von Zahlensystemen im Allgemeinen und der Obengenannten im Speziellen • rechnen mit komplexen Zahlen in Normal- und Polardarstellung und Wechseln zwischen diesen Darstellungen • berechnen lineare Abhängigkeiten, Unterräume, Basen, Skalarprodukte, Determinanten • vergleichen Lösungsmethoden zu linearen Gleichungssystemen • bestimmen Lösungen zu Eigenwertproblemen • überprüfen Eigenschaften linearer Abbildungen und Matrizen • überprüfen die Konvergenz von Zahlenfolgen • ermitteln Grenzwerte und überprüfen Stetigkeit • entwickeln Beweise anhand grundlegender Beweismethoden aus den genannten Themenbereichen • kennen eine regelmäßige selbstständige Nachbereitung und Anwendung des Vorlesungsstoffes
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 1

9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung schriftlich (90 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) schriftlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Skripte des Dozenten W. Merz, P. Knabner, Mathematik für Ingenieure und Naturwissenschaftler, Springer, 2013 Fried, Mathematik für Ingenieure I für Dummies I, Wiley A. Hoffmann, B. Marx, W. Vogt, Mathematik für Ingenieure 1, Pearson v. Finckenstein et.al: Arbeitsbuch Mathematik fuer Ingenieure: Band I Analysis und Lineare Algebra. Teubner-Verlag 2006, ISBN 9783835100343 Meyberg, K., Vachenauer, P.: Höhere Mathematik 1. 6. Auflage, Sprinbger-Verlag, Berlin, 2001

1	**Modulbezeichnung** 67640	**Mathematik für INF 2** Mathematics for INF 2	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	apl.Prof.Dr. Martin Gugat
5	**Inhalt**	*Differentialrechnung einer Veränderlichen:* Ableitung mit Rechenregeln, Mittelwertsätze, LHospital, Taylor-Formel, Kurvendiskussion *Integralrechnung einer Veränderlichen:* Riemann-Integral, Hauptsatz der Infinitesimalrechnung, Mittelwertsätze, Partialbruchzerlegung, uneigentliche Integration *Folgen und Reihen:* reelle und komplexe Zahlenfolgen, Konvergenzbegriff und -sätze, Folgen und Reihen von Funktionen, gleichmäßige Konvergenz, Potenzreihen, iterative Lösung nichtlinearer Gleichungen *Grundlagen Analysis mehrerer Veränderlicher:* Grenzwert, Stetigkeit, Differentiation, partielle Ableitungen, totale Ableitung, allgemeine Taylor-Formel
6	**Lernziele und Kompetenzen**	Die Studierenden • analysieren Funktionen einer reellen Veränderlichen mit Hilfe der Differentialrechnung • berechnen Integrale von Funktionen mit einer reellen Veränderlichen • stellen technisch-naturwissenschaftliche Problemstellungen mit mathematischen Modellen dar und lösen diese • erklären den Konvergenzbegriff bei Folgen und Reihen • berechnen Grenzwerte und rechnen mit diesen • analysieren und klassifizieren Funktionen mehrerer reeller Veränderlicher an Hand grundlegender Eigenschaften • wenden grundlegende Beweistechniken in o.g. Bereichen an • erkennen die Vorzüge einer regelmäßigen Nachbereitung und Vertiefung des Vorlesungsstoffes
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 2
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung schriftlich (90 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) schriftlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 84 h Eigenstudium: 141 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Skripte des Dozenten G. Baron und P. Kirschenhofer: Einführung in die Mathematik für Informatiker 1 und 2, Springer, 1989 und 1990 K.-H. Kiyek und F. Schwarz: Mathematik für Informatiker 1 und 2, Teubner, 1989 und 1990 K. Finck von Finckenstein, J. Lehn et. al., Arbeitsbuch für Ingenieure, Band I, Teubner M. Fried: Mathematik für Ingenieure I für Dummies. Wiley M. Fried: Mathematik für Ingenieure II für Dummies. Wiley W. Merz, P. Knabner: Mathematik für Ingenieure und Naturwissenschaftler, Springer, 2013

1	**Modulbezeichnung** 67650	**Mathematik für INF 3** Mathematics for INF 3	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Mathematik für Ingenieure C3:INF (4 SWS) Übung: IngMathC3U (2 SWS)	7,5 ECTS -
3	Lehrende	Prof. Dr. Carsten Gräser	

4	**Modulverantwortliche/r**	apl.Prof.Dr. Martin Gugat
5	**Inhalt**	*Anwendung der Differentialrechnung im R^n^ * Extremwertaufgaben, Extremwertaufgaben mit Nebenbedingungen, Lagrange-Multiplikatoren, Theorem über implizite Funktionen, Anwendungsbeispiele *Optimierung und Algebraische Strukturen* Mathematische Grundlagen der linearen Optimierung und geometrische Interpretation, Simplex, Konvexität, Dualität Binäre Operationen, Monoide, Halbgruppen, Gruppen, Homomorphismen, Ringe, Körper, Vektorräume über endlichen Körpern, Einführung in Kryptographie und Kanalcodierung *Gewöhnliche Differentialgleichungen* Explizite Lösungsmethoden, Existenz- und Eindeutungssätze, Lineare Differentialgleichungen, Systeme von Differentialgleichungen, Eigen- und Hauptwertaufgaben, Fundamentalsysteme, Stabilität
6	**Lernziele und Kompetenzen**	Die Studierenden • klassifizieren verschiedene Extremwertaufgaben anhand der Nebenbedingungen und kennen die grundlegende Existenzaussagen • erschließen den Unterschied zur eindimensionalen Kurvendiskussion, • wenden die verschiedene Extremwertaufgaben bei Funktionen mehrerer Veränderlicher mit und ohne Nebenbedingungen, • beherrschen die grundlegenden Begriffe der Algebra, • klassifizieren gewöhnliche Differentialgleichungen nach Typen, • wenden elementare Lösungsmethoden auf Anfangswertprobleme bei gewöhnlichen Differentialgleichungen an, • wenden allgemeine Existenz- und Eindeutigkeitsresultate an, • erschließen den Zusammenhang zwischen Analysis und linearer Algebra, • wenden die erlernten mathematischen Methoden auf die Ingenieurswissenschaften an, • beachten die Vorzüge einer regelmaessigen selbstaendigen Nachbereitung und Vertiefung des Vorlesungsstoffes.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 3

9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung schriftlich (90 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) schriftlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Skripte des Dozenten M. Fried, Mathematik für Ingenieure II für Dummies, Wiley A. Hoffmann, B. Marx, W. Vogt: |Mathematik für Ingenieure 1,2| Pearson K. Finck von Finckenstein, J. Lehn et. al.: |Arbeitsbuch für Ingenieure, Band I und II|, Teubner H. Heuser: |Gewöhnliche Differentialgleichungen| Teubner

1	**Modulbezeichnung** 67660	**Mathematik für INF 4** Mathematics for INF 4	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	apl.Prof.Dr. Martin Gugat
5	**Inhalt**	*Allgemeine und Diskrete Wahrscheinlichkeitsrechnung:* • Rechenregeln für Wahrscheinlichkeiten • Bedingte Wahrscheinlichkeiten, stochastische Unabhängigkeit, Produktexperimente • Charakterisierung diskreter Wahrscheinlichkeitsverteilungen durch Wahrscheinlichkeitsfunktion, erzeugende Funktion, Momente • Eigenschaften und Anwendungsgebiete der wichtigsten diskreten Verteilungen • Markoffketten *Kontinuierliche Wahrscheinlichkeitsrechnung:* • Charakterisierung von Wahrscheinlichkeitsverteilungen auf dem R^n^ durch Verteilungsfunktionen und Dichten • Verteilung, Erwartungswert und Varianz von Zufallsvariablen • Funktionen von Zufallsvariablen • Mehrdimensionale Normalverteilung • Grenzwertsätze *Statistische Datenanalyse:* • StatistischeAnalysemethoden: Parameterschätzung, Konfidenzbereiche, Signifikanztests • Lineare statistische Modelle: Regression und Varianzanalyse
6	**Lernziele und Kompetenzen**	Die Studierenden • untersuchen die grundlegende Begriffe und Methoden der diskreten und kontinuierlichen Wahrscheinlichkeitsrechnung, • berechnen charakteristische Größen (Erwartungswerte, Momente, erz. Funktion) • erstellen stochastische Modelle in typischen Anwendungssituationen, • kennen Eigenschaften von diskreten und kontinuierlichen Verteilungen und nutzen diese bei der Lösung von Problemstellungen, • beherrschen und verwenden Rechenverfahren für stochastische Fragestellungen, • untersuchen die Eigenschaften von Schätzern, • wenden Verfahren der beschreibenden und mathematischen Statistik, insbesondere Schätzer und Signifikanztests • analysieren Modelle, die auf homogenen Markoffketten basieren • schätzen die Vorzüge einer regelmäßigen Nachbereitung und Vertiefung des Vorlesungsstoffes

7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	schriftlich (90 Minuten) Übungsleistung
11	**Berechnung der Modulnote**	schriftlich (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 84 h Eigenstudium: 141 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Skripte des Dozenten • Gerhard Hübner, Stochastik, Vieweg + Teubner, 2009 • Ulrich Krengel, Einführung in die Wahrscheinlichkeitstheorie und Statistik, Vieweg Verlag • John A. Rice, Mathematical Statistics and Data Analysis, Duxburry Press

1	**Modulbezeichnung** 93040	**Parallele und Funktionale Programmierung** Parallel and functional programming	**5 ECTS**
2	Lehrveranstaltungen	Übung: PFP-R-Fr12 (2 SWS) Vorlesung: Parallele und Funktionale Programmierung (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Julian Brandner Dr.-Ing. Norbert Oster Prof. Dr. Michael Philippsen	

4	**Modulverantwortliche/r**	Dr.-Ing. Norbert Oster Prof. Dr. Michael Philippsen
5	**Inhalt**	• Grundlagen der funktionale Programmierung • Grundlagen der parallelen Programmierung • Datenstrukturen • Objektorientierung • Scala-Kenntnisse • Erweiterte JAVA-Kenntnisse • Aufwandsabschätzungen • Grundlegende Algorithmen
6	**Lernziele und Kompetenzen**	Die Studierenden • erlernen die Grundlagen der funktionalen Programmierung anhand der Programmiersprache Scala • verstehen paralleles Programmieren mit Java • kennen fundamentale Datenstrukturen und Algorithmen • können funktionale und parallele Algorithmen entwickeln und analysieren
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 3
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93150	**Rechnerkommunikation** Computer communications	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Reinhard German
5	**Inhalt**	Die Vorlesung vermittelt die Grundlagen der Rechnerkommunikation und durchläuft von oben nach unten die Schichten des Internets: • Anwendungsschicht • Transportschicht • Netzwerkschicht • Sicherungsschicht • Physikalische Schicht Sicherheit wird als übergreifender Aspekt behandelt. An verschiedenen Stellen werden analytische Modelle eingesetzt, um Wege für eine quantitative Auslegung von Kommunikationsnetzen aufzuzeigen. Die Übung beinhaltet praktische und theoretische Aufgaben zum Verständnis der einzelnen Schichten.
6	**Lernziele und Kompetenzen**	Die Studierenden erwerben • Kenntnisse über zentrale Mechanismen, Protokolle und Architekturen der Rechnerkommunikation (Topologie, Schicht, Adressierung, Wegsuche, Weiterleitung, Flusskontrolle, Überlastkontrolle, Fehlersicherung, Medienzugriff, Bitübertragung) am Beispiel des Internets und mit Ausblicken auf andere Netztechnologien • Kenntnisse über Sicherheit, Leistung und Zuverlässigkeit bei der Rechnerkommunikation • praktische Erfahrung in der Benutzung und Programmierung von Rechnernetzen
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 2
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (90 Minuten) Hausaufgaben zu Rechnerkommunikation (Übungsleistung): • Studienleistung, Übungsleistung, unbenotet, 2.5 ECTS • weitere Erläuterungen: Bearbeitung (zwei)wöchentlicher Aufgabenblätter in Gruppenarbeit. Für den unbenoteten Übungsschein sind 60% der Punkte je Aufgabenblatt zu erreichen Rechnerkommunikation (Klausur): • Prüfungsleistung, Klausur, Dauer (in Minuten): 90, benotet, 2.5 ECTS

		• Anteil an der Berechnung der Modulnote: 100.0 %
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Lehrbuch: Kurose, Ross. Computer Networking. 8th Ed., Pearson, 2021.

1	**Modulbezeichnung** 93180	**Systemprogrammierung** System programming	**10 ECTS**
2	Lehrveranstaltungen	Vorlesung: Systemprogrammierung 2 (2 SWS) Übung: SP-RÜ R14 (2 SWS) Übung: SP1-T09 (2 SWS) Vorlesung: Systemprogrammierung 1 (2 SWS) Übung: Systemprogrammierung 1 - Übungen (für Wiederholer) (2 SWS)	- - 2,5 ECTS 2,5 ECTS -
3	Lehrende	Prof. Dr.-Ing. Rüdiger Kapitza Dr.-Ing. Jürgen Kleinöder Luis Gerhorst Eva Dengler Jonas Rabenstein Maximilian Ott Signe Schwarz-Rüsch Thomas Preisner	

4	**Modulverantwortliche/r**	Dr.-Ing. Jürgen Kleinöder Prof. Dr.-Ing. Wolfgang Schröder-Preikschat
5	**Inhalt**	• Grundlagen von Betriebssystemen (Adressräume, Speicher, Dateien, Prozesse, Koordinationsmittel; Betriebsarten, Einplanung, Einlastung, Virtualisierung, Nebenläufigkeit, Koordination/Synchronisation) • Abstraktionen/Funktionen UNIX-ähnlicher Betriebssysteme • Programmierung von Systemsoftware • C, Make, UNIX-Shell (Solaris, Linux, MacOS X)
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben fundierte Kenntnisse über Grundlagen von Betriebssystemen • verstehen Zusammenhänge, die die Ausführungen von Programmen in vielschichtig organisierten Rechensystemen ermöglichen • erkennen Gemeinsamkeiten und Unterschiede zwischen realen und abstrakten (virtuellen) Maschinen • erlernen die Programmiersprache C • entwickeln Systemprogramme auf Basis der Systemaufrufschnittstelle UNIX-ähnlicher Betriebssysteme
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 2;3
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur mit MultipleChoice (120 Minuten)

11	**Berechnung der Modulnote**	Klausur mit MultipleChoice (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 180 h Eigenstudium: 120 h
14	**Dauer des Moduls**	2 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Lehrbuch: Betriebssysteme Grundlagen, Entwurf, Implementierung, Wolfgang Schröder-Preikschat, 2008

1	**Modulbezeichnung** 93121	**Theorie der Programmierung** Theory of programming	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Lutz Schröder
5	**Inhalt**	• Termersetzungssysteme, Normalisierung, Konfluenz • Getypter und ungetypter Lambda-Kalkül • Semantik von Programmiersprachen, Anfänge der Bereichstheorie • Datentypen, Kodatentypen, Induktion und Koinduktion, Rekursion und Korekursion • Programmverifikation, Floyd-Hoare-Kalkül • Reguläre Sprachen und endliche Automaten • Beschriftete Transitionssysteme, Bisimulation und Temporallogik
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben elementare Definitionen und Fakten zu den behandelten Formalismen wieder. Verstehen Die Studierenden • erläutern Grundbegriffe der Syntax und Semantik von Formalismen und setzen diese zueinander in Bezug • beschreiben und erklären grundlegende Algorithmen zu logischem Schließen und Normalisierung • beschreiben wichtige Konstruktionen von Modellen, Automaten und Sprachen Anwenden Die Studierenden • verfassen formale Spezifikationen sequentieller und nebenläufiger Programme • verifizieren einfache Programme gegenüber ihrer Spezifikation durch Anwendung der relevanten Kalküle • ° führen einfache Beweise über Programme mittels Induktion und Koinduktion Analysieren Die Studierenden wählen für gegebene Verifikationsprobleme geeignete Formalismen aus erstellen einfache Meta-Analysen formaler Systeme, etwa Konfluenzprüfung von Termersetzungssystemen - führen einfache Meta-Beweise über Formalismen mittels Induktion und Koinduktion Lern- bzw. Methodenkompetenz Die Studierenden beherrschen das grundsätzliche Konzept des Beweises als hauptsächliche Methode des Erkenntnisgewinns in der theoretischen Informatik.

		Sie überblicken abstrakte Begriffsarchitekturen. Sozialkompetenz Die Studierenden lösen abstrakte Probleme in kollaborativer Gruppenarbeit.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Pflichtmodul Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Glynn Winskel, Formal Semantics of Programming Languages, MIT Press, 1993 • Michael Huth, Mark Ryan, Logic in Computer Science, Cambridge University Press, 2. Auflage 2004 • Henk Barendregt, The lambda-Calculus: Its Syntax and Semantics, North Holland, 1984 • John E. Hopcroft, Jeffrey D. Ullman and Rajeev Motwani, Introduction to Automata Theory, Languages, and Computation, 3rd ed., Prentice Hall, 2006 • Franz Baader, Tobias Nipkow, Term Rewriting and All That, Cambridge University Press, 1999

Nebenfach

1	**Modulbezeichnung** 1818	**Nebenfach Physik** Minor subject: Physics	**15 ECTS**
2	Lehrveranstaltungen	Übung: Übungen zur Experimentalphysik für Nanotechnologen I (1 SWS) Vorlesung: Experimentalphysik für Materialwissenschaftler, Nanotechnologen und Integrated Life Scientists I (3 SWS) Übung: Übungen zur Experimentalphysik für Integrated Life Scientists I (1 SWS) Übung: Übungen zur Experimentalphysik für Naturwissenschaftler I (1 SWS) Vorlesung: Experimentalphysik für Naturwissenschaftler I (4 SWS)	- 5 ECTS - - 5 ECTS
3	Lehrende	Prof. Dr. Alexander Schneider Prof. Dr. Norbert Lindlein	

4	**Modulverantwortliche/r**	Prof. Dr. Vojislav Krstic Prof. Dr. Alexander Schneider Prof. Dr. Heiko Weber
5	**Inhalt**	*Mechanik:* • Messungen, Einheiten, Dimensionen, Größenordungen • Bewegungen in einer Raumdimension • Bewegungen in drei Raumdimensionen • Newtonsche Gesetze: Kraft • Arbeit, Energie, Leistung • Schwerpunkt, Impuls, Stoßprozesse • Drehbewegungen • Gravitationsgesetz • Mechanik deformierbarer Körper, Flüssigkeiten, Gase *Schwingungen und Wellen:* • ungedämpfte, gedämpfte sowie erzwungene Schwingungen • Überlagerung • Wellenausbreitung • Beugung • geometrische Optik *Thermodynamik:* • Temperatur, ideales Gas • Kinetische Gastheorie • Reales Gas, Phasendiagramm • Wärmekapazität, Schmelz-, Verdampfungsenergie • Wärmeleitung, Wärmestrahlung • Wärmekraftmaschinen, Wirkungsgrad
6	**Lernziele und Kompetenzen**	Die Studierenden • können die Grundlagen der Mechanik und Thermodynamik darstellen • haben ein grundlegendes Verständnis, wie Naturvorgänge auf grundlegende Naturgesetze zurückgeführt werden können

		• wenden in Übungen das erlernte Wissen auf spezielle Situationen und Fragestellungen der Mechanik und Thermodynamik an • besitzen grundlegende Kompetenz im analytischen Denken als Mittel zur exakten Beschreibung naturwissenschaftlicher Zusammenhänge
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Leistungsschein schriftlich (90 Minuten) schriftlich (90 Minuten) Klausur (180 Minuten) schriftlich (180 Minuten) schriftlich schriftlich (90 Minuten) schriftlich (90 Minuten) Klausur (90 Minuten) Klausur (90 Minuten) schriftlich (180 Minuten) Variabel schriftlich
11	**Berechnung der Modulnote**	Leistungsschein (0%) schriftlich (33%) schriftlich (33%) Klausur (100%) schriftlich (67%) schriftlich (33%) schriftlich (33%) schriftlich (33%) Klausur (33%) Klausur (33%) schriftlich (67%) Variabel (100%) schriftlich (33%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Wiederholung der Prüfungen**	Die Prüfungen dieses Moduls können nur einmal wiederholt werden.
14	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
15	**Dauer des Moduls**	1 Semester
16	**Unterrichts- und Prüfungssprache**	Deutsch

17	**Literaturhinweise**	Paul A. Tipler: Physik, Spektrum Akademischer Verlag (2009) Hering, Martin, Stohrer: Physik für Ingenieure, Springer (2012) Gerthsen: Physik, Springer (2010)

1	**Modulbezeichnung** 1821	**Nebenfach Soziologie** Minor subject: Sociology	**15 ECTS**
2	Lehrveranstaltungen	Seminar: Basisseminar: Vergleichende Gesellschaftsanalyse (2 SWS) Seminar: Basisseminar: Bildung und Lebenslauf (2 SWS) Vorlesung: Vorlesung: Einführung in die Methoden der empirischen Sozialforschung (2 SWS) Seminar: Basisseminar: Vergleichende Gesellschaftsanalyse (2 SWS) Seminar: Basisseminar: Arbeit und Organisation (2 SWS, WiSe 2023) Seminar: Basisseminar: Kultur und Kommunikation (2 SWS, WiSe 2023) Seminar: Basisseminar: Bildung und Lebenslauf (2 SWS) Vorlesung: Vorlesung: Einführung in die Soziologie (2 SWS) Seminar: Basisseminar: Bildung und Lebenslauf (2 SWS, WiSe 2023) Seminar: Basisseminar: Arbeit und Organisation (2 SWS, WiSe 2023) Seminar: Basisseminar Arbeit und Organisation (0 SWS) Seminar: Basisseminar Vergleichende Gesellschaftsanalyse (SWS) Seminar: Basisseminar: Kultur und Kommunikation (SWS, WiSe 2023)	- - - - - - - 5 ECTS - - - - -
3	Lehrende	Amelie Tihlarik Dr. Manuel Nicklich Prof. Dr. Nicole Janine Saam Philipp Landauer Dr. Michael Whittall Prof. Dr. Aida Bosch Jakob Seitz PD Dr. Gerd Sebald Tobias Schramm Dr. Ronald Staples PD Dr. Gerd Sebald Dr. Alexander Antony	

4	**Modulverantwortliche/r**	PD Dr. Gerd Sebald
5	**Inhalt**	Im WiSe: Vorlesung Einführung in die Soziologie, dazu ein Seminar aus den vier Gegenstandsfeldern in der Soziologie mit 5 ECTSIm SoSe:

		Vorlesung Einführung in die Methoden der empirischen Sozialforschung (das Seminar kann auch im SoSe absolviert werden)
6	**Lernziele und Kompetenzen**	Siehe Veranstaltungen
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Nebenfach Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten) schriftlich/mündlich schriftlich/mündlich Klausur (60 Minuten) schriftlich/mündlich Variabel schriftlich/mündlich
11	**Berechnung der Modulnote**	Klausur (33%) schriftlich/mündlich (33%) schriftlich/mündlich (33%) Klausur (33%) schriftlich/mündlich (33%) Variabel (100%) schriftlich/mündlich (33%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 360 h
14	**Dauer des Moduls**	2 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Nebenfach Astronomie

1	**Modulbezeichnung** 66292	**Astronomie** Astronomy	**15 ECTS**
2	Lehrveranstaltungen	Übung: Übung zur Einführung in die Astronomie 1 (1 SWS, WiSe 2023) Vorlesung: Einführung in die Astronomie 1 (2 SWS) Praktikum: Astronomisches Praktikum (LAG) (8 SWS) Vorlesung: Einführung in die Astronomie 2 (2 SWS) Praktikum: Astronomisches Praktikum (Informatiker) (8 SWS) Übung: Übung zur Einführung in die Astronomie 2 (1 SWS) Praktikum: Astronomisches Praktikum (Nicht-Physiker) (8 SWS)	- 3 ECTS - - - - -
3	Lehrende		

4	**Modulverantwortliche/r**	Prof. Dr. Ulrich Heber Prof. Dr. Manami Sasaki Prof. Dr. Jörn Wilms
5	**Inhalt**	Das Modul gibt eine Beschreibung der wesentlichen Bestandteile des Universums und der naturwissenschaftlichen Methoden, die es uns erlauben, ihre Entfernungen, Größenskalen, Massen und physikalische Natur zu verstehen. Im Einzelnen werden behandelt: • Geschichtlicher Hintergrund der Astronomie • Sonnensystem: Planetenbewegung und Keplersche Gesetze, Eigenschaften der Planeten und der kleinen Objekte im Sonnensystem (Auswahl aus: innerer Aufbau der Planeten, planetare Oberflächen, Atmosphären, Ringe), extrasolare Planeten. • Sterne: Entfernungen, Temperaturen, Spektren, Massen, Hertzsprung-Russell-Diagramm, innerer Aufbau, Entstehung und Entwicklung, Endstadien der Sternentwicklung, Doppelsterne. • Milchstraße und andere Galaxien: Aufbau und Entwicklung, Klassifikation, kosmischer Materiekreislauf, Galaxienhaufen, ausgewählte Methoden der Entfernungsbestimmung. • Das Universum: Entstehung, Hubblesches Gesetz, 3K Hintergrundstrahlung, Entwicklung des Universums. • Astronomische Messmethoden: Aufbau und Benutzung astronomischer Teleskope, Spektroskopie, Detektoren
6	**Lernziele und Kompetenzen**	Die Studierenden • berichten über die wichtigsten Bestandteile des Universums und ihrer Entwicklung. • erläutern Methoden zur Messung der Entfernungen von Sternen und Galaxien und wenden diese auf Messungen an.

		• bestimmen aus Messdaten Massen und Temperaturen astronomischer Objekte. • führen einfache astronomische Messungen selbst durch und werten die Ergebnisse aus. • beschreiben die in der Astronomie notwendige Extrapolation von Ergebnissen von Labormessungen auf astronomische Skalen. • bedienen typische astronomische Instrumente.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Astronomie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel Klausur schriftlich/mündlich Protokollheft
11	**Berechnung der Modulnote**	Variabel (33%) Klausur (67%) schriftlich/mündlich (33%) Protokollheft (0%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 165 h Eigenstudium: 285 h
14	**Dauer des Moduls**	2 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• H. Karttunen, P. Kroger, H. Oja, |Fundamental Astronomy|, Springer, 2003 • M. Kutner, |Astronomy: A Physical Perspective|, Cambridge Univ. Press, 2003

Nebenfach Betriebswirtschaftslehre

1	**Modulbezeichnung** 83467	**Business Process Management**	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Martin Matzner Dr. Sven Weinzierl
5	**Inhalt**	• Einführung in Business Process Management • Der BPM-Lebenszyklus • Prozessidentifikation: Kontext, Prozessarchitekturen, Auswahl / Priorisierung von Prozessen zur Optimierung • Einführung in Prozessmodellierung mit BPMN • Fortgeschrittene Prozessmodellierung: Wiederholungen, Nachbesserungen, Ereignisse, Ausnahmen, Regeln, Best Practices • Prozessentdeckung: Methoden, Modellierung, Qualitätskontrolle • Qualitative Prozessanalyse • Quantitative Prozessanalyse • Prozess-Redesign: Hintergründe, Transaktionale Methoden, Transformative Methoden • Prozessgewahre Informationssysteme: Arten, Vorteile, Herausforderungen • Prozessimplementierung mit ausführbaren Modellen • Prozessüberwachung: Kontext und Ansätze, Techniken aus dem Process Mining, Performancemessung, • Techniken für Geschäftsprozessmanagement in wissensintensiven Prozessen • Business Process Management als Unternehmensfähigkeit
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben fundierte Kenntnisse über Grundfragen, Begrifflichkeit und praktische Relevanz des Geschäftsprozessmanagements, • können zentrale Konzepte in der Prozessmodellierung und -automatisierung verstehen und erklären, • können verschiedene Arten von Modellierungsnotationen (imperativ, deklarativ) unterscheiden und erklären, • können verschiedene Stufen im BPM-Lebenszyklus und deren Anforderungen an Stakeholder verstehen und erklären, • sind in der Lage, Geschäftsprozesse in BPMN zu verstehen und zu modellieren, • sind in der Lage, Geschäftsprozesse zu analysieren und optimieren, • sind in der Lage, aus den umfangreichen Techniken, Notationen und Konzepten aus dem Business Process Management für den jeweiligen Einsatzbereich anwendbare zu wählen und einzusetzen.

7	**Voraussetzungen für die Teilnahme**	Keine.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Betriebswirtschaftslehre Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2021). *Grundlagen des geschäftsprozessmanagements: Übersetzt von Thomas Grisold, Steven Groß, Jan Mendling, Bastian Wurm*. Springer Berlin Heidelberg.

1	**Modulbezeichnung** 82570	**BWL für Ingenieure** Business studies for engineers	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: BWL für Ingenieure I (2 SWS) Vorlesung mit Übung: BWL für Ingenieure II (2 SWS)	- -
3	Lehrende	Lars Friedrich Prof. Dr. Kai-Ingo Voigt	

4	**Modulverantwortliche/r**	Prof. Dr. Kai-Ingo Voigt
5	**Inhalt**	BW 1 (konstitutive Grundlagen): Grundlagen und Vertiefung spezifischer Aspekte der Rechtsform-, Standort-, Organisations- und Strategiewahl BW 2 (operative Leistungsprozesse): Betrachtung der unternehmerischen Kernprozesse Forschung und Entwicklung mit Fokus auf das Technologie- und Innovationsmanagement, Beschaffung und Produktion sowie Marketing und Vertrieb BW 3 (Unternehmensgründung): Grundlagen der Gründungsplanung und des Gründungsmanagements BW 3 Übung (Vertiefung und Businessplanerstellung): Vertiefung einzelner Schwerpunkte aus den Bereichen BW 1, 2 und 3 sowie ausgewählte Fallstudien zu wichtigen Elementen eines Businessplans
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben Kenntnisse über Grundfragen der allgemeinen Betriebswirtschaftslehre • verstehen die Kernprozesse der Unternehmung und die damit verbundenen zentralen Fragestellungen • erwerben ein Verständnis für den Entwicklungsprozess der Unternehmung sowie deren Kernprozesse, insbesondere verfügen sie über breites und integriertes Wissen einschließlich der wissenschaftlichen Grundlagen in den Bereichen Forschung und Entwicklung, Beschaffung, Produktion, Marketing und Vertrieb. • können Fragen des Technologie- und Innovationsmanagements anhand der Anwednung ausgewählter Methoden und Instrumente erschließen • wissen um die Bestandteile eines Businessplans, deren Bedeutung und sind in der Lage, diese zu verfassen und zu beurteilen
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Betriebswirtschaftslehre Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	2 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Voigt, Industrielles Management, 2008

1	**Modulbezeichnung** 82397	**E-Business und E-Commerce** E-Business and E-Commerce	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Verena Tiefenbeck
5	**Inhalt**	Die Digitalisierung nimmt einen immer größeren Teil der Geschäftswelt ein. Die elektronische Unterstützung, Abwicklung und Aufrechterhaltung von Geschäftsprozessen wird als E-Business bezeichnet. Einen Teilbereich dessen macht der elektronische Austausch von Gütern und Dienstleistungen, der sog. E-Commerce, aus. Die Veranstaltung beleuchtet die Konzepte E-Business und E-Commerce tiefergehend entlang der gesamten Wertschöpfungskette. Dabei werden zunächst Entwicklungen betrachtet, die die Verbreitung von E-Business und E-Commerce ermöglicht haben. Darüber hinaus werden Besonderheiten digitaler Geschäftsmodelle herausgearbeitet und analogen Ansätzen gegenübergestellt. Des Weiteren erfolgt eine kritische Einordung der sich daraus ergebenden Chancen und Risiken. Die verschiedenen Aspekte werden anhand aktueller Praxisbeispiele verdeutlicht.
6	**Lernziele und Kompetenzen**	Die Studierenden sind in der Lage, ... • die Konzepte E-Business und E-Commerce voneinander abzugrenzen • zentrale zugrundliegende betriebswirtschaftliche Konzepte, Technologien und Standards zu nennen und zu erklären • Besonderheiten digitaler Geschäftsmodelle zu erklären • Beispiele für digitale Geschäftsmodelle und die IT-Unterstützung entlang der Wertschöpfungskette zu identifizieren und kritisch zu beurteilen • Chancen und Herausforderungen bei der Entwicklung und Einführung von E-Business-Anwendungen zu benennen und bei der Analyse von Praxisbeispielen zu bewerten • Das Wechselspiel zwischen BWL und Informations- und Kommunikationstechnologien anhand konkreter Beispiele zu erklären • In Gruppenarbeit unterschiedliche Aspekte des E-Business und E-Commerce anhand konkreter Fallstudien herauszuarbeiten, zu präsentieren und im Plenum zu diskutieren
7	**Voraussetzungen für die Teilnahme**	keine
8	**Einpassung in Studienverlaufsplan**	Semester: 2;4

9	**Verwendbarkeit des Moduls**	Nebenfach Betriebswirtschaftslehre Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Referat (20 Minuten) Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Referat (30%) Klausur (70%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Kurspaket mit Lehrmaterial und Literatur

1	**Modulbezeichnung** 86960	**Enterprise Content and Collaboration Management** Enterprise content und collaboration management	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Enterprise Content and Collaboration Management (4 SWS)	5 ECTS
3	Lehrende	Prof. Dr. Sven Laumer Tim-Julian Schwehn	

4	**Modulverantwortliche/r**	Prof. Dr. Sven Laumer
5	**Inhalt**	Das Modul bietet eine Einführung in Konzepte und Strategien des Enterprise Content und Collaboration Managements sowie in Technologien, Werkzeuge und Methoden, die verwendet werden, um Wissens- und Informationsflüsse in Unternehmen zu organisieren. Die Vorlesung fokussiert auf die Digitalisierung und neue Formen der Arbeit. Hierzu werden in der Veranstaltung theoretische und technische Grundlagen von digitalen Arbeitsgruppen, digitalen Gemeinschaften und dem Management von digitalen Inhalten (Content, Informationen, Wissen) vermittelt. Der Fokus liegt darauf, wie Arbeit in Teams und Arbeitsgruppen organisiert werden muss und wie digitale Technologien (z.B. Social-Media-Anwendungen) gestaltet sein müssen, um diese Abläufe effektiv und effizient zu unterstützen. Die Übung fokussiert sich auf konkrete digitale Technologien und deren Anwendung, um Informations- und Wissensflüsse in Unternehmen zu unterstützen. In rechnergestützten Übungen werden grundlegende Funktionen verschiedener ECM-Systeme vorgestellt und von den Studenten am Rechner vertieft.
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben ein grundlegendes Verständnis der Rolle des Enterprise Content Management in der Unternehmenspraxis • kennen die Funktionalitäten und Merkmale von ECM-Systemen • sind in der Lage, Nutzungsszenarien von ECM in Unternehmen zu analysieren und zu konzipieren • können dank der erfolgten Rechnerübungen ein ECM-System auf verschiedenen Plattformen (u.a. Microsoft SharePoint) in seinen Grundfunktionen konfigurieren
7	**Voraussetzungen für die Teilnahme**	keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Betriebswirtschaftslehre Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Alalwan, J. A. (2012): Enterprise content management research: a comprehensive review. In: Journal of Enterprise Information Management 25 (5), pp. 441461. Laumer, S., Maier, C., and Weitzel, T. (2015) Successfully Implementing Enterprise Content Management: Lessons Learnt from a Financial Service Provider Proceedings of the 36th International Conference on Information Systems (ICIS), Fort Worth, TX, USA. Laumer, S., Beimborn, D., Maier, C., and Weinert, C. (2013) Enterprise Content Management, Business & Information Systems Engineering (BISE) (5:6), p. 449-452. Simons, A., and vom Brocke, J. (2014): "Enterprise content management in information systems research." Enterprise Content Management in Information Systems Research. Springer, Berlin, Heidelberg. Tyrväinen, P.; Päivärinta, T.; Salminen, A., and Iivari, J. (2006): Characterizing the evolving research on enterprise content management. In: European Journal of Information Systems 15 (6), pp. 627634.

1	**Modulbezeichnung** 82491	**Marketing-Fallstudien** Marketing case studies	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Marketing Fallstudien (4 SWS)	5 ECTS
3	Lehrende	Dr. Christian Götz	

4	**Modulverantwortliche/r**	Dr. Christian Götz
5	**Inhalt**	• Kommunikation und Kommunikationsmodelle • Einführung in das moderne Marketingmanagement • Grundlagen der Marketinginstrumente • Aspekte der Marktsegmentierung und Produktplatzierung • Marktforschung und ihre Methoden • Kundenprädispositionen und Kaufbereitschaft • Kommunikationsstrategie und Werbemanagement • Grundlegende Instrumente im Kommunikationsmix • Grundzüge des Neuromarketings und der Werbepsychologie
6	**Lernziele und Kompetenzen**	Die Studierenden • erläutern die wesentlichen Inhalte und Methoden der modernen Marketinginstrumente im Marketing-Mix (Produkt-, Kontrahierungs-, Distributions- und Kommunikationspolitik) • beschreiben die wesentlichen Designs und Methoden (Beobachtung, Befragung, Experiment) der modernen Marktforschung • erklären den Prozess der Entwicklung von zielgruppenspezifischen Werbebotschaften und Aussagen • kennen wesentliche Komponenten der modernen Werbepsychologie • können diese in bestimmten Fallbeispielen (z.B. bei der Anordnung und dem Aufbau des Point of Sale) verifizieren • wissen über die Wirkung von Werbebotschaften und können diese an Beispielen analysieren • können Erkenntnisse aus der Vorlesung auf ihr eigenes On- und Offline-Einkaufsverhalten anwenden
7	**Voraussetzungen für die Teilnahme**	Grundlagenkenntnisse in BWL empfohlen aber nicht zwingend erforderlich
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Betriebswirtschaftslehre Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Präsentation/Hausarbeit
11	**Berechnung der Modulnote**	Präsentation/Hausarbeit (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Homburg, Christian: Marketingmanagement • Meffert, Heribert: Marketing • Kotler, Philip: Marketing Management • Moser, Klaus: Markt- und Werbepsychologie • Scheier, Christian, Held, Dirk: Wie Werbung wirkt. Erkenntnisse des Neuromarketings

1	**Modulbezeichnung** 82455	**Service Management und Service Engineering** Service management and service engineering	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Martin Matzner
5	**Inhalt**	Die Veranstaltung soll einen Überblick über Methoden und Modellen zur Entwicklung, zum Management und zur Erbringung von Dienstleistungen sowie Einsicht in Grundkonzepte des Forschungsgebiets der Service Science geben. Darüber hinaus werden aktuelle Trends IT-gestützter Dienstleistungen vorgestellt.
6	**Lernziele und Kompetenzen**	Die Studierenden • beherrschen Grundkonzepte der Dienstleistungsforschung, • verstehen die Bedeutung von IT-Artefakten für das Dienstleistungsmanagement, • können Methoden und Modelle des Service Engineering zur Gestaltung von Geschäftsmodellen, Erhebung von Anforderungen, Erforschung von Prozessen, und Planung von Marketing-Konzepten anwenden, • können Methoden und Modelle des Service Management zur Messung der Dienstleistungsqualität anwenden und • lernen aktuelle Anwendungsbereiche der Dienstleistungsforschung und -praxis kennen (zum Beispiel digitale Plattformen und intelligente Dienstleistungen).
7	**Voraussetzungen für die Teilnahme**	Erfolgreiches Absolvieren der Assessmentphase.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Betriebswirtschaftslehre Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Literaturverweise und Downloadmaterial im StudOn-Kurs (Link wird auf der Lehrstuhl-Website bekanntgegeben: https://www.is.rw.fau.de/lehre/veranstaltungen/service-management-und-service-engineering/).

Nebenfach Biologie

1	**Modulbezeichnung** 62921	**Allgemeine Biologie I** General biology I	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Allgemeine Biologie I: Biologie für Nebenfächler (Bio-NF) (4 SWS)	-
3	Lehrende	Dr. Gerald Seidel PD Dr. Michael Lebert Dr. Ingrid Brehm	

4	**Modulverantwortliche/r**	PD Dr. Michael Lebert
5	**Inhalt**	*Botanik:* • Bau und Leistung der Pflanzenzelle • Morphologie und Anatomie der Pflanzenorgane • Systematik und Evolution von Pflanzen • Vermehrung von Pflanzen • Pflanzenphysiologie • Pflanze und Umwelt *Zoologie:* • Stoffwechsel, Kreislauf und Atmung • erregbare Zellen: Muskelzellen und Nervenzellen • zelluläre Neurophysiologie (Ruhepotential, Aktionspotential, axonale Weiterleitung der Erregung, Synapse) *Mikrobiologie:* • Einführung in die Mikrobiologie • Zellstruktur und Zellfunktion • Grundlagen der Molekularbiologie und Bakteriengenetik • Mikrobiologie der Prokaryoten (Physiologie, Taxonomie und Phylogenie) • Grundlagen der Virologie
6	**Lernziele und Kompetenzen**	Die Studierenden • können die Struktur und Funktionen der Biomoleküle in Ihren Grundzügen beschreiben und erläutern; • verstehen die Zelltypen verschiedener Organismen und können deren Zellbestandteile- und bausteine darstellen und erklären; • kennen die Grundbegriffe der Zytologie, Morphologie und Anatomie der Pflanzen und sind in der Lage diese Einordnungen anzuwenden; • sind in der Lage, die Physiologie der Pflanzen darzustellen; • können die Anpassungen von Pflanzen darlegen; • sind befähigt, die Evolution der Pflanzen in den Grundzügen zu erklären; • können zelluläre Unterschiede zwischen Pflanzen und Tieren erläutern; • sind in der Lage, die fundamentalen Prozesse des Energiestoffwechsels der Tiere - und damit verbundene Anpassungen (Kreislauf und Atmung) in den Grundzügen darzustellen und zu beschreiben;

		• verstehen die zellulären und molekularen Grundlagen der Muskelkontraktion und können diese darstellen und verdeutlichen; • können zelluläre Grundlagen sowie grundlegende Funktionsmechanismen von Nervenzellen einordnen • verstehen den Einfluss von Mikroorganismen auf Ökosysteme und deren Nutzung in Landwirtschaft, Biotechnik, Medizin und Lebensmittelproduktion; • erwerben basale Kenntnisse der Bakteriengenetik, der Physiologie, der taxonomischer Einteilung und den Grundlagen der Virologie.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Biologie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Nultsch, Allgemeine Botanik, Thieme Verlag Fuchs, Allgemeine Mikrobiologie Thieme-Verlag Wehner, Gehring, Kühn, Zoologie, Thieme Brock: Mikrobiologie, Pearson Verlag Campbell, Biologie, Pearson

1	**Modulbezeichnung** 62922	**Allgemeine Biologie II** General biology II	**5 ECTS**
2	Lehrveranstaltungen	Übung: Allgemeine Biologie II: Morphologie und Anatomie der Organismen (Laborübung) (5 SWS)	5 ECTS
3	Lehrende	Prof. Dr. Alexandra Schambony Dr. Peter Richter PD Dr. Michael Lebert	

4	**Modulverantwortliche/r**	PD Dr. Michael Lebert
5	**Inhalt**	• Einführung in die Lichtmikroskopie (Hellfeld-, Dunkelfeld-, Phasenkontrast-Mikroskopie) • Charakteristika eukaryontischer Zellen am Beispiel von Amöben und Ciliaten (u.a. Phagocytose, verschiedene Fortbewegungstypen) • Entwicklung eines Tieres am Wirbeltierbeispiel (Huhn) • Organisationsprinzipien vielzelliger Tiere am Beispiel repräsentativer Tiergruppen (Cnidaria, Plathelminthes, Annelida, Arthropoda, Vertebrata) • Evolutive Abwandlung und ökologische Anpassungen dieser Baupläne • Algen und Pflanzen: u.a. Cyanobakterien, Kieselalgen und Grünalgen (Organisationsstufen), Moose und Farne (Aufbau und Generationswechsel), Höhere Pflanzen (Wurzel und Physiologie der Wasseraufnahme, Spross mit Leitgeweben und sekundärem Dickenwachstum, Blatt und Photosynthese, Blüte, Fortpflanzung und Frucht)
6	**Lernziele und Kompetenzen**	Die Studierenden • lernen grundlegende Mikroskopier- und Präparationstechniken • sind in der Lage mikroskopische und anatomische Präparate zeichnerisch zu protokollieren • erkennen typische tierische Gewebe in histologischen Präparaten und - kennen die charakteristischen Phasen der Entwicklung eines Wirbeltieres und können die dabei ablaufenden Prozesse wiedergeben • verstehen die basalen Funktionen wichtiger tierischer Organsysteme und können diese in den verschiedenen Bauplänen miteinander vergleichen • kennen die grundsätzlichen Trends der Evolution pflanzlicher und tierischer Baupläne und können deren adaptive Bedeutung ermessen • bekommen ein vertieftes Verständnis von anatomischen und zellulären Funktionsbeziehungen bei Pflanzen und Tieren • verfügen über Grundlegende Kenntnisse der Formenkunde
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!

9	**Verwendbarkeit des Moduls**	Nebenfach Biologie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (45 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 75 h Eigenstudium: 75 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Campbell: Biologie; Wehner/Gehring: Zoologie

1	**Modulbezeichnung** 62982	**Mikrobiologische Übungen** Microbiology: practical exercises	**5 ECTS**
2	Lehrveranstaltungen	Übung: Praktische mikrobiologische Grundlagen für Biotechnologie 1 (3 SWS) Übung: Mikrobiologische Übungen für Naturwissenschaftler (3 SWS)	- -
3	Lehrende	Dr. Gerald Seidel	

4	**Modulverantwortliche/r**	Dr. Gerald Seidel
5	**Inhalt**	Mikroskop, Färbetechniken, Kultur- und Sterilisationsverfahren, • Wachstum von Bakterien, Antibiotika • Transformation von Acinetobacter spec., • Identifizierung/Diagnostik von Bakterien • grundlegende Techniken der Molekularbiologie • Experimente: Beobachtung von Bakterien im Mikroskop, verschiedene Darstellungsverfahren • Nachweis von Keimen in der Luft • Erlernen verschiedener Techniken, Herstellung von Nährmedien, Bestimmung Zellzahl in einer Kolonie, Bestimmung der Phagenzahl in einem Plaque, Sterilisationsversuche • selektive Anreicherung von Bakterien, Bakterienwuchskurve; Einfluss von Antibiotika auf das Wachstum von Bakterien • Isolierung von Antibiotika-Produzenten • Nachweis und Identifizierung von Bakterien, Resistenzbestimmung, Isolierung von Antibiotika-Produzenten,
6	**Lernziele und Kompetenzen**	Aneignung der Grundkenntnisse der Mikrobiologie und molekularbiologischen Grundtechniken
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Biologie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	schriftlich
11	**Berechnung der Modulnote**	schriftlich (0%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 60 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch

16	**Literaturhinweise**	• Lehrbuch: Brock Mikrobiologie, M. T. Madigan & J. M. Martinko, aktuelle Ausgabe (z.Zt. 2013) • Lehrbuch: Allgemeine Mikrobiologie, G. Fuchs, aktuelle Ausgabe (z.Zt. 2007) • Lehrbuch: Mikrobiologische Methoden, E.Bast

1	**Modulbezeichnung** 62820	**Ökologie und Diversität B** Ecology and diversity B	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten. Die Übungen sind anwesenheitspflichtig.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	PD Dr. Ruth Stadler
5	**Inhalt**	**Zoologische Bestimmungsübungen**: • Erkundung typischer Biotoptypen in der Umgebung Erlangens (Kiefernwald, Weiher incl. Plankton- und Saprobienthematik, Wiese, Fließwasser etc.) und ihrer Tiergemeinschaften und Ökologie • Einführung in Präparation von Wirbellosen und Anlegen einer wissenschaftlichen zoologischen Sammlung **Botanische Bestimmungsübungen**: Erkundung von Beispielarten in der Umgebung Erlangens an folgenden Standorten: • Rathsberg: Laubmischwald: Caryophyllaceae: Stellaria, Ranunculaceae: Anemone, Violaceae: Viola, Liliaceae: Polygonatum • Regnitztal: Auwald: Brassicaceae: Alliari, Lamiaceae: Lamium, Salicaceae: Salix • Schwabachtal: Sandmagerrasen: Fabaceae: Cytisus, Rosaceae: Potentilla, Euphorbiaceae: Euphorbia • Regnitzwiesen: Kräuter der Fettwiese: Apiaceae: Anthriscus, Asteroideae: Leucanthemum, Cichorioideae: Taraxacum, Polygonaceae: Rumex • Regnitzwiesen: Gräser der Fettwiese: Poaceae: Arrhenatherum, Poa, Lolium, Festuca • Walberla: Kalkmagerrasen: Plantaginaceae: Veronica, Plantago, Orobanchaceae: Rhinanthus • Tennenlohe: Sandäcker: Chenopodiaceae: Chenopodium, Geraniaceae: Erodium • Heusteg: Verlandungsreihe eutropher Gewässer: Cyperaceae: Carex, Solanaceae: Solanum, Juncaceae: Juncus, Primulaceae: Lysimachia • An verschiedenen Standorten: sandige, nährstoffreiche Ruderalfluren: Hypericaceae: Hypericum Onagraceae: Oenothera • Reichswald: Nadelforst auf Sandböden: Ericaceae: Vaccinium, Gymnospermae: Pinus, Pteridophyta: Dryopteris
6	**Lernziele und Kompetenzen**	Die Studierenden • können die wichtigsten einheimischen Tier- und Pflanzenfamilien und deren typischer Vertreter an ihrem Standort (Exkursionen) erkennen und unterscheiden (Formenkenntnis) sowie nach Art bestimmen;

		• sind in der Lage, fachgerecht mit einem Bestimmungsschlüssel umzugehen; • sind fähig, ein wissenschaftliches Herbar und eine zoologische Sammlung anzulegen; • sind zur Teamarbeit befähigt.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Biologie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (45 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%) PL: E-Klausur im Antwortwahlverfahren 45 Min.
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 75 h Eigenstudium: 75 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Zoologie, nur Empfehlung: Brohmer: Fauna von Deutschland (Quelle und Meyer Bestimmungsbücher); Botanik: Schmeil-Fitschen: Flora von Deutschland (Quelle und Meyer Bestimmungsbücher); Rothmaler: Exkursionsflora (Springer); Oberdorfer: Pfl.-soziol. Exkursionsflora (Ulmer)

1	**Modulbezeichnung** 62940	**Zoologie** Zoology	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Ökologie und Diversität A: Vorlesung Einführung in die Ökologie, Zoologie und Botanik (2 SWS) Übung: Ökologie und Diversität A: Zoologische und botanische Bestimmungsübungen (3 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Dr. Jürgen Schmidl PD Dr. Ruth Stadler Dr. Ulrike Daigl	

4	**Modulverantwortliche/r**	Dr. Jürgen Schmidl
5	**Inhalt**	*Vorlesung*: Grundkenntnisse zu folgenden Teilgebieten der Ökologie, Zoologie und Botanik: • Morphologie (Systematik des Tier- und Pflanzenreiches, Kennenlernen ausgewählter Baupläne und Taxa) • Evolution (Grundlagen, Mechanismen und ökologische Aspekte der Evolution) • Phylogenie (Methoden der Systematik und Taxonomie, Artkonzepte, Stammbäume) • Ökologie (Grundlagen, Großlebensräume/Ökosysteme der Erde, Einnischung von Tier-/Pflanzenarten, Aut-, Dem- und Synökologie, Makroökologie, Muster und Prozesse, Diversität) • Biogeographie (Konzepte und geologisch-historische Grundlagen der globalen Verbreitung der Tier- und Pflanzengruppen) *Übungen*: • Einführung in die Arbeit mit dem Stereomikroskop. • Morphologie, Systematik und Diversität wichtiger heimischer Tier- und Pflanzengruppen und ihrer typischen Vertreter • Übungen zum Bestimmen heimischer Arten mittels Bestimmungsschlüssel und elektronischer Medien • Biologie und Ökologie der bestimmten Arten und Gruppen
6	**Lernziele und Kompetenzen**	Die Studierenden • können wichtige Tier und Pflanzengruppen unterscheiden und typische Vertreter erkennen; • verstehen die Diversität im Tier- und Pflanzenreich; • können die Grundlagen der Morphologie, Evolution, Phylogenie, Ökologie und Biogeographie darstellen und erklären; • sind befähigt zum Erkennen und Lösen von relevanten Problemen aus systematisch-ökologischen Teilgebieten der Zoologie und Botanik; • sind in der Lage, mit Bestimmungsschlüsseln und einschlägigen Medien umzugehen; • sind fähig, die Vorlesungsinhalte in Übungen praktisch umzusetzen;

		• haben den fachgerechten Umgang mit dem Stereomikroskop vermittelt bekommen.
7	**Voraussetzungen für die Teilnahme**	keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Biologie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (45 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 75 h Eigenstudium: 75 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Nur Empfehlung: • Brohmer: Fauna von Deutschland (Quelle und Meyer Bestimmungsbücher); • Schmeil-Fitschen: Die Flora Deutschlands • Wehner/Gehring: Zoologie (Thieme); • Strasburger: Lehrbuch der Pflanzenwissenschaften (Springer)

Nebenfach Chemie

1	**Modulbezeichnung** 62963	**Allgemeine Organische Chemie** General organic chemistry	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Andreas Hirsch
5	**Inhalt**	Grundlegende Konzepte und Stoffklassen der Organischen Chemie, chemische Terminologie, chemische Eigenschaften: • Alkane, Alkene, Alkine • Delokalisierte pi-Systeme • Stereoisomerie • Alkohole • Ether • Aldehyde und Ketone • Carbonsäuren und Derivate • Amine und Aminosäuren • Heterocylcen • Dicarbonylverbindungen • Biopolymere und Bioaggregate - Grundbausteine des Lebens und der Biochemie • Biochemische Grundprozesse
6	**Lernziele und Kompetenzen**	Die Studierenden • kennen die elementaren Stoffklassen organischer Moleküle und können deren physikalische und chemische Eigenschaften verstehen und einschätzen • kennen die Eigenschaften von funktionellen Gruppen in organischen Molekülen • beherrschen die chemische Terminologie und einfache Syntheseprinzipien.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Chemie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 75 h Eigenstudium: 75 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	K.P.C Vollhardt, N.E. Schore: Organische Chemie (WileyVCH)

1	**Modulbezeichnung** 62060	**Allgemeine und Anorganische Chemie** Lecture general and inorganic chemistry	**10 ECTS**
2	Lehrveranstaltungen	Praktikum: Anorganisch-chemisches Praktikum für Nebenfachstudierende (7 SWS) Vorlesung: Allgemeine und Anorganische Chemie (mit Experimenten) (4 SWS, WiSe 2023)	- -
3	Lehrende	Dr. Jörg Sutter Prof. Dr. Sjoerd Harder Dr. Jens Langer Prof. Dr. Karsten Meyer	

4	**Modulverantwortliche/r**	Prof. Dr. Sjoerd Harder
5	**Inhalt**	*Grundzüge der Allgemeinen und Anorganischen Chemie:* Atommodelle, Aufbau des Periodensystems, chemische Bindungsarten, grundlegende anorganische Verbindungsklassen, Gasgesetze, Stöchiometrie, chemisches Rechnen, Zustandsdiagramme, chemische Thermodynamik und Kinetik, Theorie des Übergangszustandes, Katalyse, chemisches Gleichgewicht, Redox-Reaktionen, Säure/Base-Reaktionen, Elektrolyse/Galvanisches Element, Chemie der Elemente (Hauptgruppenelemente), Grundlagen der Koordinationschemie *Spektroskopische Methoden* für kinetische, mechanistische und strukturelle Untersuchungen *Kurspraktikum:* Umgang mit anorganischen Säuren und Basen, Salzen und Komplexverbindungen, Grundzüge der qualitativen chemischen Analytik durch einfache Versuche mit Basisverbindungen der anorganischen Chemie, nasschemische Nachweise für Metall-Kationen und Anionen Einführung in sicheres Arbeiten mit Gefahrstoffen in chemischen Laboratorien; Umgang mit chemischen Abfällen
6	**Lernziele und Kompetenzen**	Die Studierenden • erläutern die Grundlagen der allgemeinen und anorganischen Chemie • wenden spektroskopische Methoden für kinetische, mechanistische und strukturelle Untersuchungen an • setzen die Vorlesungsinhalte im Kurspraktikum um und führen die im Praktikumsplan vorgesehenen Versuche selbständig durch • kennen den Umgang mit Gefahrstoffen und Abfällen in chemischen Laboratorien
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Chemie Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	Klausur (45 Minuten) Praktikumsleistung
11	**Berechnung der Modulnote**	Klausur (100%) Praktikumsleistung (0%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 180 h Eigenstudium: 120 h
14	**Dauer des Moduls**	2 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• C.E. Mortimer, |Chemie das Basiswissen der Chemie|, Georg Thieme Verlag • E. Riedel, |Anorganische Chemie|, de Gruyter • C. E. Housecroft, A.G. Sharpe, |Anorganische Chemie|, Pearson • E. Dane, F. Wille, H. Laatsch: Kleines Chemisches Praktikum, 10. Auflage, Wiley-VCH, Weinheim 2004

Nebenfach Englische Linguistik

1	**Modulbezeichnung** 77011	**Basismodul II Linguistics** Basic module II: Linguistics	**10 ECTS**
2	Lehrveranstaltungen	Aufbauseminar: AS English Linguistics II (Erlangen) (2 SWS) Grundseminar: English Linguistics I (2 SWS) Vorlesung: Basisvorlesung English Linguistics II (1 SWS) Aufbauseminar: Aufbauseminar English Linguistics II (Erlangen) (2 SWS) Vorlesung: Basisvorlesung: Principles of English Linguistics (SWS, WiSe 2023)	- - - - -
3	Lehrende	Annika Kaßler Armine Garibyan Leonarda Prela Hendrik Kligge Richenda Wright Panagiotis Kenanidis Bastian Führer Thomas Maisel Dr. Michael Klotz	

4	**Modulverantwortliche/r**	Armine Garibyan Dr. Peter Uhrig
5	**Inhalt**	Im Basismodul werden die Studierenden mit folgenden linguistischen Konzepten und Ansätzen vertraut gemacht: • Linguistik als deskriptive Disziplin • Prinzipien und Methodik der modernen Linguistik • Phonetik und Phonologie • Syntax und Grammatik • Morphologie, Wortbildung, Lexikologie und Phraseologie • Semantik • Pragmatik • Regionale und historische Varietäten • Kontrastive Linguistik • Lexikografie Soweit es in diesem Basismodul möglich ist, werden den Studierenden unterschiedliche Ansätze innerhalb der verschiedenen linguistischen Teilbereiche präsentiert. Die Inhalte der Basisvorlesung sind für die Klausur des Aufbauseminars relevant.
6	**Lernziele und Kompetenzen**	Die Studierenden nennen und erläutern grundlegende linguistische Konzepte und ihr erlangtes Wissen um das Wesen von Sprache im Allgemeinen und des Englischen im Besonderen. Sie beginnen, die Methodik der linguistischen Analyse zu erfassen und zu beschreiben. Im ersten Semester fassen die Studierenden Konzepte und Ansätze aus der Perspektive eines als Standardmodell anerkannten linguistischen Modells zusammen und erläutern diese. Durch die gemeinsame Arbeit

		an Sprachmaterial, die sich über den gesamten Semesterverlauf zieht, erarbeiten sich Studierende im gegenseitigen, diskursiven Austausch ein grundständiges Wissen um grundlegende linguistische Begrifflichkeit und die kritische Methode, die sie im Seminarverbund anwenden. Im zweiten Semester kontrastieren die Studierenden dann diese Beschreibungen mit alternativen Theorien und Ansätzen. Sie verstehen den Umgang mit Termini und Methoden auch im Kontext von unterschiedlichen linguistischen Theoriemodellen und hinterfragen verschiedene Ansätze, wobei sie zusätzlich kritisch argumentieren. Die Einführung in linguistische Disziplinen wie Soziolinguistik und historische Linguistik dient außerdem dazu, dass Studierende grundlegende Fakten über den historischen und kulturellen Hintergrund englischsprachiger Länder, vor allem Großbritanniens und der USA angeben und erläutern können. Auf diese Weise betten sie linguistische Konzeptionen in einen größeren soziokulturellen Kontext ein.
7	**Voraussetzungen für die Teilnahme**	Keine; in der Regel soll das Grundseminar vor dem Aufbauseminar und der Basisvorlesung absolviert werden. Da die Inhalte der Basisvorlesung für die Klausur des Aufbauseminars relevant sind, sollten beide Veranstaltungen im gleichen Semester besucht werden.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Englische Linguistik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (beste Note%) Klausur (beste Note%)
12	**Turnus des Angebots**	jedes 2. Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	2 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Veranstaltungsspezifisch - wird zu Beginn des Semesters bekanntgegeben.

1	**Modulbezeichnung** 77230	**Seminarmodul L-UF Linguistics** Teaching seminar module: Linguistics	**5 ECTS**
2	Lehrveranstaltungen	Mittelseminar: Teaching Grammar for Grammar Haters (2 SWS)	-
3	Lehrende	Tobias Wiedenmann Dr. Brigitta Mittmann Thomas Maisel	

4	**Modulverantwortliche/r**	Dr. Peter Uhrig
5	**Inhalt**	Im Seminarmodul L-UF Linguistics beschäftigen sich die Studierenden intensiv mit einer linguistischen Teildisziplin. Nach einer Wiederholung grundlegender Modelle, die für diese Teildisziplin relevant sind, vertiefen die Studierenden ihre Kenntnisse, indem sie unterschiedliche Bereiche und Ansätze der Teildisziplin kennenlernen und diskutieren. Darüber hinaus wählen die Studierenden einen thematischen Schwerpunkt, den sie selbstständig bearbeiten. Linguistische Teildisziplinen, die in diesem Modul behandelt werden, können eher theoretischer Natur (z.B. Syntax, Phonetik / Phonologie, Semantik) oder angewandter Natur (z.B. Lexikografie, Korpuslinguistik, Linguistics and Language Teaching) sein.
6	**Lernziele und Kompetenzen**	Im Seminarmodul L-UF Linguistics werden die theoretischen Ansätze, die die Studierenden im Elementarmodul kennengelernt haben vertieft und auf eine höhere Reflexionsstufe gestellt. Im diskursiven Austausch mit anderen Seminarteilnehmenden erwerben und vertiefen die Studierenden dabei die Fähigkeit, wissenschaftlich begründete Positionen argumentativ zu vertreten und kritisch zu reflektieren. Die Studierenden lernen, unterschiedliche linguistische Modelle und Lösungsansätze zu diskutieren und evaluieren. Darüber hinaus beschäftigen sich die Studierenden erstmals eigenständig mit der Deskription und Analyse linguistischer Daten. Auf der Ebene der *Study Skills* verbessern die Studierenden ihre Fähigkeiten im Erarbeiten und Halten mündlicher Präsentationen. Weiterhin werden die Studierenden auf das Verfassen einer schriftlichen Hausarbeit vorbereitet, wobei insbesondere Aspekte wie Umgang mit wissenschaftlichen Quellen, sprachlicher Ausdruck und Eingrenzung eines Themas im Vordergrund stehen. Eine Adressaten-, situations- und themenadäquate Textkompetenz wird erlangt, sowie das Geben und Verarbeiten von Feedback aus der Gruppe (*peer review*).
7	**Voraussetzungen für die Teilnahme**	Erfolgreiche Teilnahme an den Elementarmodulen L-UF Linguistics I und II
8	**Einpassung in Studienverlaufsplan**	Semester: 2
9	**Verwendbarkeit des Moduls**	Nebenfach Englische Linguistik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	schriftlich

11	**Berechnung der Modulnote**	schriftlich (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

Nebenfach Geowissenschaften

1	**Modulbezeichnung** 64935	**Rohstoffe und Nachhaltigkeit** Raw materials and sustainability	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Karsten Haase
5	**Inhalt**	In unserer auf Technologie basierenden Gesellschaft ist die nachhaltige Nutzung von Ressourcen der Erde ein brisantes Thema und wird es voraussichtlich auch auf längere Sicht bleiben, da die Wirtschaft der Bundesrepublik Deutschland vom Import vieler Rohstoffe abhängig ist. So erfordert z.B. der Umbau zu einer Kohlenstoffarmen Energieerzeugung mit dem Ausbau von Stromversorgung und Elektromobilität gewaltige Mengen von Metallen wie Kupfer oder Kobalt. In unserem im Sinne einer Bildung für nachhaltigen Entwicklung konzipierten Seminar Rohstoffe und Nachhaltigkeit werden die Herausforderung der Gewinnung, Nutzung und Aufbereitung von Ressourcen unter dem Aspekt der Nachhaltigkeit vermittelt und gemeinsam diskutiert. Fokus sind neben den Prozessen der unterschiedlichen Lagerstättenbildungen und dem Einfluss ihrer Nutzung auf die Umwelt und den Menschen die gesellschaftspolitischen, wirtschaftlichen und rechtlichen Fragestellungen und Herausforderungen die dieses Thema aufwirft. Zu den Themen halten Expert*innen aus den Geowissenschaften, der Chemie, Materialwissenschaft, der Politik und der Wirtschaft Vorträge, die dann in einem blended Learning Format diskutiert und besprochen werden. Dieses interdisziplinäre Seminar hat damit auch das Ziel, gemeinsam Lösungswege hin zu einer nachhaltigen Gesellschaft auszuloten.
6	**Lernziele und Kompetenzen**	Die Studierenden können • die Bildung, Nutzung und Aufbereitung von Lagerstätten unter dem Aspekt der Nachhaltigkeit beschreiben, erklären und diskutieren • moderene Möglichkeiten des Recyclings von Rohstoffen erklären und diskutieren • wirtschaftliche und politische Zusammenhänge und Abhängigkeiten in Dtl. von Rohstoffen erklären • Nachhaltige Aspekte im Bezug auf Rohstoffe (kritische Metalle und Wasser) diskutieren
7	**Voraussetzungen für die Teilnahme**	Keine; Link zum StudOn Kurs: https://www.studon.fau.de/studon/goto.php?target=crs_4314344
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Geowissenschaften Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung

11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Wiederholung der Prüfungen**	Die Prüfungen dieses Moduls können nur einmal wiederholt werden.
14	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 21 h Eigenstudium: 129 h
15	**Dauer des Moduls**	1 Semester
16	**Unterrichts- und Prüfungssprache**	Englisch
17	**Literaturhinweise**	Reader wird vom Lehrstuhl nach Anmeldung bereitgestellt

1	**Modulbezeichnung** 64980	**Z-Edu-Geo 1 - Das System Erde** Z-Edu-Geo 1 - Earth as a system	**5 ECTS**
2	Lehrveranstaltungen	Seminar: Kompetenzseminar zum Klimawandel (insbesondere für Lehramtsstudierende und Nebenfachstudierende) (2 SWS) Seminar: [Z-Edu-Geo1] + [Nebenfach] + [Modulstudien Naturale] Das System Erde (2 SWS)	5 ECTS -
3	Lehrende	Katrin Schwarzfischer Julia Gebert Laura Heinl Dr. Anette Regelous	

4	**Modulverantwortliche/r**	Dr. Anette Regelous
5	**Inhalt**	Die Inhalte des Seminars sind: • Entstehung der Elemente • Entstehung des Sonnensystems • Aufbau und Entwicklung des Systems Erde • Grundlagen des Vulkanismus und der Plattentektonik • Einführung in die Seismik • Entstehung und Entwicklung der kontinentalen und ozeanischen Kruste • Übersicht über die Entwicklung des Lebens und der Atmosphäre • Grundlagen über die wichtigsten Minerale und Gesteine • Einführung in die Bildung von Rohstoffen • Entstehung der Atmosphäre • Entwicklung des Klimas durch die Erdgeschichte
6	**Lernziele und Kompetenzen**	Die Studierenden können • die Entstehung der Elemente erläutern • fachwissenschaftliche Grundlagen und die Zusammenhänge des System Erde erklären • die Entstehung und Entwicklung der ozeanischen und kontinentalen Kruste wiedergeben • Grundlagen der Forschungsmethodik wie z.B. Seismik erklären • die Entwicklung des Lebens und der Atmosphäre erläutern • Ursachen und Auswirkungen des Klimawandels wiedergeben
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Geowissenschaften Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Präsentation (60 Minuten)

11	**Berechnung der Modulnote**	Präsentation (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 75 h Eigenstudium: 75 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 64985	**Z-Edu-Geo 2 - Der Mensch im geowissenschaftlichen Umfeld** Z-Edu-Geo 2 - Humans in the geoscientific environment	**5 ECTS**
2	Lehrveranstaltungen	Seminar: [Z - Edu - Geo 2] + [Mathe + Informatik] + [SQ] Der Mensch im geol. Umfeld (2 SWS) Übung: Archäometrie [Z-Edu-Geo 2] + [SQ] + [Werkstoffwiss] (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr. Matthias Göbbels Andreas Murgan	

4	**Modulverantwortliche/r**	Prof. Dr. Matthias Göbbels
5	**Inhalt**	Die Inhalte des Seminars Der Mensch im geowissenschaftlichen Umfeld" sind fachliche Grundlagen über die Mensch-Umwelt-Beziehungen im geologischen Kontext, wie z.B. • Grundlagen im geologischen Kontext zu Mensch-Umwelt-Beziehung • Geothermie • Hydrogeologie • Rohstoffe • Glas & Keramik Die Inhalte der Übung sind praktische Anwendungen und Anschauungsmaterial begleitend zum Seminar. In der Veranstaltung Archäometrie werden Materialien und Techniken vom Altertum bis zur Neuzeit im technisch-gesellschaftlichen Kontext vorgestellt und bewertet. Dabei sind z.B. Kupfer, Bronze, Eisen und Edelmetalle wichtige Themen.
6	**Lernziele und Kompetenzen**	Die Studierenden können • Grundlagen der Mensch-Umwelt-Beziehung erläutern und anwenden • Beispiele für technische Anwendung in der Geologie anführen und beschreiben • Geologische Herausforderungen der Gesellschaft diskutieren und bewerten • Materialien und Techniken im historisch-gesellschaftlichen Kontext in der Entwicklung nachvollziehen und verstehen
7	**Voraussetzungen für die Teilnahme**	keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Geowissenschaften Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich (30 Minuten)
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Werden von den Dozenten in den jeweiligen Veranstaltungen vorgestellt.

Nebenfach Germanistische Linguistik

1	**Modulbezeichnung** 77301	**Grundlagen der Germanistischen Linguistik 1 (Ling BM 1)** Foundations of German linguistics I	**5 ECTS**
2	Lehrveranstaltungen	**Ling BM-1** - Vorlesung: Sichere Systeme (2 SWS, WiSe 2023) Einführungskurs: Kurs 2 (Fr-Rädle) (4 SWS) Tutorium: Ling BM-1: Tutorium (ER-Abbass) (1 SWS) Einführungskurs: Kurs 3 (Di-NM-Durst) (3 SWS) Tutorium: Ling BM-1: Tutorium (Nürnberg - Korbinian Schwarz) (1 SWS)	 2,5 ECTS 5 ECTS - 5 ECTS -
3	Lehrende	Prof. Dr.-Ing. Felix Freiling Dr. Karin Rädle Uwe Durst Christine-Maria Coca Wiebke Blanck	

4	**Modulverantwortliche/r**	Prof. Dr. Christian Merkl Dr. Karin Rädle Prof. Dr. Johannes Rincke Prof. Dr. Regina Therese Riphahn
5	**Inhalt**	• Vermittlung von Grundlagen zu zentralen Teilbereichen der Sprachwissenschaft • Einführung in die grundlegende Fachterminologie der germanistischen Linguistik • Darstellung der zentralen Hilfsmittel und Arbeitsmethoden • Einführung in problemorientierte Fragestellungen • Einführung in Grundlagen der Sprachanalyse Das Einführungsseminar Grundlagen der Sprachwissenschaft • bietet einen Überblick über die linguistischen Teilgebiete Zeichentheorie, Phonetik/Phonologie, Graphematik/ Orthographie, Morphologie, Wortbildung, Syntax, Semantik und Pragmatik, • führt in die zentralen sprachwissenschaftlichen Methoden ein, • vermittelt einen Überblick über Forschungsbereiche, die auf Aspekte der Sprachverwendung bezogen sind Es bleibt vorbehalten, dass Teile des Einführungsseminars im Plenum abgehalten werden. Das Tutorium dient der Vertiefung und Übung der im Modul gebotenen Kenntnisse und Methoden. Hinweis: Für das Erreichen der Modulziele – insbesondere für das Einüben und Verfestigen von Analysefähigkeiten, die interaktive Wissensvermittlung und die praxisorientierte Förderung spezifischer germanistischer Kompetenzen – ist eine aktive Mitarbeit der Studierenden unerlässlich.

6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben grundlegende Kenntnisse in den wesentlichen Methoden und Arbeitsmitteln, • können die vorgestellten Theorien und Methoden kritisch reflektieren, • lernen, die Sprache auf verschiedenen sprachstrukturellen Ebenen zu unterscheiden, und • sind in der Lage, sprachliche Ebenen in Ansätzen zu analysieren.
7	**Voraussetzungen für die Teilnahme**	a) für den Studiengang BA Germanistik: keine b) für das Lehramt (vertieft/nicht vertieft studiert): keine c) für weitere Studiengänge: keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Germanistische Linguistik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 80 h Eigenstudium: 70 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 77432	**Vertiefungsmodul Gegenwartssprache/DAF (Ling VM 2)** Specialisation module: Contemporary German/German as a foreign language	**10 ECTS**
2	Lehrveranstaltungen	Kolleg: Kolleg: Grundlagen des Erwerbs und der Vermittlung des Deutschen als Fremd- und Zweitsprache (ER) (2 SWS) Hauptseminar: Ling-HS: Methoden in der Linguistik und in der Lexikographie (Schierholz) (2 SWS) Hauptseminar: Ling-HS: Die Wortstellung des Deutschen (Paranhos-Zitterbart) (2 SWS) Kolleg: Kolleg: Grundlagen des Erwerbs und der Vermittlung des Deutschen als Fremd- und Zweitsprache (Nürnberg) (2 SWS) Es besteht Anwesenheitspflicht. Die Fähigkeiten und Kompetenzen werden in der gemeinsamen Diskussion entwickelt; die Diskursivierung des Wissens ist ein zentraler performativer Bestandteil des Seminars. Da sich Hauptseminar oder Kolleg als Spezialveranstaltung verstehen, sind die Inhalte untrennbar an die Person des Lehrenden gebunden; es ist daher nicht möglich, den Besuch der Lehrveranstaltung durch Selbststudium zu kompensieren.	- - - -
3	Lehrende	Prof. Dr. Eva Breindl-Hiller Prof. Dr. Stefan Schierholz Dr. Christine Ganslmayer Dr. Jussara Paranhos Zitterbart Dr. Karin Rädle	

4	**Modulverantwortliche/r**	Prof. Dr. Eva Breindl-Hiller Prof. Dr. Mechthild Habermann Prof. Dr. Peter Otto Müller
5	**Inhalt**	• Vermittlung von vertieften Kenntnissen zu spezielleren, ausgewählten Themen aus den Bereichen der Grammatik des Deutschen (Phonologie, Graphematik, Morphologie, Syntax, Textlinguistik) oder zu Semantik und Lexikon der deutschen Gegenwartsprache oder zu Deutsch als Fremdsprache • Analyse und Beschreibung themenbezogener zentraler sprachlicher Erscheinungen • Diskussion themenbezogener theoretischer Konzepte Das Hauptseminar (HS) behandelt ein spezielles, weiterführendes Thema aus den Bereichen Grammatik / Textlinguistik des Deutschen, Semantik und Lexikon der deutschen Gegenwartssprache oder Deutsch als Fremdsprache.

		Das Kolleg behandelt ein spezielles, weiterführendes Thema aus den Bereichen Grammatik / Textlinguistik des Deutschen, Semantik und Lexikon der deutschen Gegenwartssprache oder Deutsch als Fremdsprache. Hinweis: Für das Erreichen der Modulziele – insbesondere für das Einüben und Verfestigen von Analysefähigkeiten, die interaktive Wissensvermittlung und die praxisorientierte Förderung spezifischer germanistischer Kompetenzen – ist eine aktive Mitarbeit der Studierenden unerlässlich.
6	**Lernziele und Kompetenzen**	Die Studierenden • erlangen einen tieferen Einblick in die Strukturiertheit exemplarischer Teilbereiche der deutschen Grammatik/ Textlinguistik oder des deutschen Wortschatzes in seiner gegenwartssprachlichen Dimension oder in den Bereich Deutsch als Fremdsprache, • gewinnen Vertrautheit in der Anwendung linguistischer Methoden und • erwerben eine vertiefte Analysekompetenz durch eigenständige Auseinandersetzung mit themenbezogenen wissenschaftlichen Fragestellungen
7	**Voraussetzungen für die Teilnahme**	a) für den Studiengang BA Germanistik: Empfehlung: alle Basismodule und Aufbaumodule [Ling BM 1, Ling BM 2, Ling AM] b) für das Lehramt (vertieft/nicht vertieft studiert): Empfehlung: alle Basismodule und Aufbaumodule [Ling BM 1, Ling BM 2, Ling AM] c) für weitere Studiengänge: Empfehlung: Basismodule [Ling BM 1, Ling BM 2]
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Germanistische Linguistik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Referat und Hausarbeit
11	**Berechnung der Modulnote**	Referat und Hausarbeit (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Altmayer, Claus/Biebighäuser, Katrin/Haberzettl, Stephanie/Heine, Antje (Hg.)(2021): Handbuch Deutsch als Fremd- und Zweitsprache. Kontexte - Themen - Methoden. Stuttgart: Metzler.

Literatur zur Vorbereitung:

Albert, Ruth/ Marx, Nicole (2016): Empirisches Arbeiten in Linguistik und Sprachlehrforschung. Anleitung zu quantitativen Studien von der Planungsphase bis zum Forschungsbericht. 3., überarb. und aktual. Aufl. Tübingen.

Engelberg, Stefan/ Lemnitzer, Lothar (2009): Lexikographie und Wörterbuchbenutzung. 4., überarb. und erw. Aufl. Tübingen.

Mann, Michael/ Schierholz, Stefan J. (2014): Methoden in der Lexikographie und Wörterbuchforschung. Ein Überblick mit einer Auswahlbibliographie. In: Lexicographica 30: 3-57.

Hennig, Mathilde (Hg.): Linguistische Komplexität – ein Phantom? Tübingen: Stauffenburg. S. 1-18.)
Settinieri, Julia/Demirkaya, Sevilen/Feldmeier, Alexis/Gültekin - Karakoç, Nazan/Riemer, Claudia (Hg.) (2014): Empirische Forschungsmethoden für Deutsch als Fremd- und Zweitsprache. Paderborn: Schöningh.

Duden (2021). Sprachliche Zweifelsfälle. Das Wörterbuch für richtiges und gutes Deutsch. 9., überarbeitete u. erweiterte Auflage. Herausgegeben von Mathilde Hennig. Berlin: Dudenverlag (= Duden Bd. 9).

Klein, Wolf Peter (2018): Sprachliche Zweifelsfälle im Deutschen. Theorie, Praxis, Geschichte. Berlin/ Boston: de Gruyter (de Gruyter Studium).
Einführende Lektüre:

		Duden (2016): Die Grammatik. Unentbehrlich für richtiges Deutsch. 9., vollständig überarbeitete und aktualisierte Auflage. (= Duden. Bd. 4). Herausgegeben von der Dudenredaktion. Mannheim [etc.]. Duden (2022): Die Grammatik: Struktur und Verwendung der deutschen Sprache. Sätze - Wortgruppen – Wörter. 10., völlig neu verfasste Auflage. (= Duden. Bd. 4). Herausgegeben von der Dudenredaktion. Mannheim [etc.]. Altmayer, Claus/Biebighäuser, Katrin/Haberzettl, Stephanie/Heine, Antje (Hg.)(2021): Handbuch Deutsch als Fremd- und Zweitsprache. Kontexte - Themen - Methoden. Stuttgart: Metzler.

1	**Modulbezeichnung** 77402	**Vertiefungsmodul Sprachwandel und Variation (Ling VM 1)** Specialisation module: Language change and variation	**10 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten. Es besteht Anwesenheitspflicht. Die Fähigkeiten und Kompetenzen werden in der gemeinsamen Diskussion entwickelt; die Diskursivierung des Wissens ist ein zentraler performativer Bestandteil des Seminars. Da sich Hauptseminar oder Kolleg als Spezialveranstaltung verstehen, sind die Inhalte untrennbar an die Person des Lehrenden gebunden; es ist daher nicht möglich, den Besuch der Lehrveranstaltung durch Selbststudium zu kompensieren.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Mechthild Habermann
5	**Inhalt**	• Darstellung von Sprache in ihrer sozialen, räumlichen, situationsbezogenen und historischen Dimension • Analyse und Beschreibung von sprachlichen Varietäten älterer Sprachstufen des Deutschen, von regionalen Varietäten, von Fach-, Sonder- und Gruppensprachen oder von Formen sprachlichen Handelns (Pragmatik, Gesprächslinguistik) im Allgemeinen • Analyse und Beschreibung zentraler Sprachwandelerscheinungen des Deutschen Das Hauptseminar behandelt ein spezielles, weiterführendes Thema aus den Bereichen Historische Linguistik, Sprachwandel, Variations-, Sozio- und Pragmalinguistik. Das Kolleg behandelt ein spezielles, weiterführendes Thema aus den Bereichen Historische Linguistik, Sprachwandel, Variations-, Sozio- und Pragmalinguistik. Hinweis: Für das Erreichen der Modulziele – insbesondere für das Einüben und Verfestigen von Analysefähigkeiten, die interaktive Wissensvermittlung und die praxisorientierte Förderung spezifischer germanistischer Kompetenzen – ist eine aktive Mitarbeit der Studierenden unerlässlich.
6	**Lernziele und Kompetenzen**	Die Studierenden • erhalten Einblick in die Vielfalt sprachlicher Variationen und Sprachkontakte unterschiedlicher Provenienz im soziokulturellen Kontext in Geschichte und Gegenwart sowie in zentrale Bereiche des Sprachwandels, • entwickeln ein Verständnis für Sprachen als historisch gewordene, identitätsstiftende Einheiten,

		• erwerben Beschreibungs- und Erklärungskompetenzen für die Erfassung sprachlicher Varianten und ihrer soziokulturellen Bedingtheiten in Geschichte und Gegenwart • erwerben grundlegende methodische Kompetenzen im Umgang mit Sprachkorpora.
7	**Voraussetzungen für die Teilnahme**	a) für den Studiengang BA Germanistik: Empfehlung: alle Basismodule und Aufbaumodule [Ling BM 1, Ling BM 2, Ling AM] b) für das Lehramt (vertieft/nicht vertieft studiert): Empfehlung: alle Basismodule und Aufbaumodule [Ling BM 1, Ling BM 2, Ling AM] c) für weitere Studiengänge: Empfehlung: Basismodule [Ling BM 1, Ling BM 2]
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Germanistische Linguistik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Referat und Hausarbeit
11	**Berechnung der Modulnote**	Referat und Hausarbeit (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Nebenfach Mathematik

1	**Modulbezeichnung** 65311	**Algebra**	**10 ECTS**
2	Lehrveranstaltungen	Tutorium: Tutorium zur Algebra (2 SWS) Vorlesung: Algebra (4 SWS) Übung: Übungen zur Algebra (2 SWS)	- - -
3	Lehrende	Prof. Dr. Wolfgang Ruppert	

4	**Modulverantwortliche/r**	Prof. Dr. Friedrich Knop
5	**Inhalt**	• Gruppentheorie: Untergruppen, Quotienten, Operationen von Gruppen, endlich erzeugte abelsche Gruppen • Ringtheorie: Ideale, Quotienten, Polynomringe, maximale Ideale, • Irreduzibilität • Elementare Zahlentheorie: Restklassenringe, Eulersche phi-Funktion, Chinesischer Restsatz, quadratisches Reziprozitätsgesetz Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • nennen und erklären algebraische Strukturen anhand von Gruppen, Ringen und Körpern und verwenden diese; • behandeln auch komplexe Symmetrien mittels Gruppentheorie selbständig; • lösen geometrische und zahlentheoretische Probleme mittels Ringtheorie und Zahlentheorie; • sammeln und bewerten relevante Informationen und erkennen Zusammenhänge.
7	**Voraussetzungen für die Teilnahme**	empfohlen: Module Lineare Algebra I und II
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur mit Übungsleistung
11	**Berechnung der Modulnote**	Klausur mit Übungsleistung (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 105 h Eigenstudium: 195 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch

16	**Literaturhinweise**	• M. Artin: Algebra • Fischer: Algebra • N. Jacobson: Basic Algebra I, II + Skript • S. Lang: Algebra

1	**Modulbezeichnung** 65092	**Computational complexity**	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Ioannis Giannakopoulos
5	**Inhalt**	Potential topics include: • P, NP, and NP-completeness • Complexity classes and reductions • Boolean circuits • The polynomial-time hierarchy • Space complexity • Randomized computation • Counting complexity • Introduction to the PCP theorem and hardness of approximation • Average-case complexity
6	**Lernziele und Kompetenzen**	Upon successful completion of the module, students: • Have a rigorous understand of the concept of computation and its formal limitations • Have knowledge of the fundamental complexity classes (including P, NP and PSPACE) • Understand the notion of completeness and are able to design and understand reductions between these classes • Are exposed to various formal computation models, including Boolean circuits and randomness
7	**Voraussetzungen für die Teilnahme**	• Undergraduate-level course in discrete mathematics (including, e.g. basics of combinatorics, graphs, logic and set theory) • Undergraduate-level course in algorithms and/or discrete optimization • Basic knowledge of analysis, linear algebra and probability
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich Oral exam (30 minutes)
11	**Berechnung der Modulnote**	mündlich (100%) Oral exam (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 45 h Eigenstudium: 105 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Sanjeev Arora and Boaz Barak. “Computational Complexity: A Modern Approach”. Cambridge University Press, 2009 • Oded Goldreich. “Computational Complexity: A Conceptual Perspective”. Cambridge University Press, 2008. • Christos H. Papadimitriou. “Computational Complexity”. Addison-Wesley, 1994.

1	**Modulbezeichnung** 65917	**Discrete optimization I**	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Discrete Optimization I (2 SWS) Übung: Tutorial zu Discrete Optimization I (1 SWS)	5 ECTS -
3	Lehrende	Prof. Dr. Frauke Liers-Bergmann Florian Rösel	

4	**Modulverantwortliche/r**	Prof. Dr. Frauke Liers-Bergmann
5	**Inhalt**	Theoretical and practical fundamentals of solving difficult mixed-integer linear optimization problems (MIPs) constitute the main focus of this lecture. At first, the concept of NP-completeness and a selection of common NP-complete problems will be presented. As for polyhedral theory, fundamentals concerning the structure of faces of convex polyhedra will be covered. Building upon these fundamentals, cutting plane algorithms as well as branch-and-cut algorithms for solving MIPs will be taught. Finally, some typical problems of discrete optimization, e.g., the knapsack problem, the traveling salesman problem or the set packing problem will be discussed.
6	**Lernziele und Kompetenzen**	Students • will gain basic theoretical knowledge of solving mixed-integer linear optimization problems (MIPs), • are able to solve MIPs with the help of state-of-the-art optimization software.
7	**Voraussetzungen für die Teilnahme**	Recommended: Linear and Combinatorial Optimization
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 45 h Eigenstudium: 105 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 65933	**Discrete optimization II**	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Timm Oertel
5	**Inhalt**	In this lecture, we cover theoretical aspects and solution strategies for difficult integer and mixed-integer optimization problems. First, we show the equivalence between separation and optimization. Then, we present solution strategies for large-scale optimization problems, e.g., decomposition methods and approximation algorithms. Finally, we deal with conditions for the existence of integer polyhedra. We also discuss applications for example from the fields of engineering, finance, energy or public transport.
6	**Lernziele und Kompetenzen**	Students • use basic terms of discrete optimization • model real-world discrete optimization problems, determine their complexity and solve them with appropriate mathematical methods.
7	**Voraussetzungen für die Teilnahme**	Recommended: Knowledge in linear and combinatorial optimization, discrete optimization I
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 45 h Eigenstudium: 105 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Lecture notes • Bertsimas, Weismantel: Optimization over Integers, Dynamic Ideas, 2005 • Conforti, Cornuéjols, Zambelli: Integer Programming, Springer 2014 • Nemhauser, Wolsey: Integer and Combinatorial Optimization, Wiley 1994 • Schrijver: Combinatorial optimization Vol. A-C, Springer 2003

		• Schrijver: Theory of Linear and Integer Programming, Wiley, 1986 • Wolsey: Integer Programming, Wiley, 2021

1	**Modulbezeichnung** 65231	**Diskretisierung und numerische Optimierung** Discretisation and numerical optimisation	**10 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Eberhard Bänsch
5	**Inhalt**	Teil 1: Diskretisierung Ein- und Mehrschrittverfahren für Anfangswertaufgaben gewöhnlicher Differentialgleichungen: • explizite und implizite Runge-Kutta-Verfahren, BDF, Extrapolation • asymptotische Stabilität (Nullstabilität), Konsistenz, Konvergenz • Steifheit und Stabilität bei fester Schrittweite • Schrittweiten- und Ordnungsadaptivität • Randwertaufgaben für gewöhnliche Differentialgleichungen • Einführung in Finite-Element-Verfahren Teil 2: Unrestringierte Optimierung • Abstiegsverfahren • CG-Verfahren (mit Vorkonditionierung, CG-Newton) • Quadratische Optimierungsprobleme • Penalty- und Barriereverfahren Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • verwenden algorithmische Zugänge zu Problemen, die mittels gewöhnlicher Differentialgleichungen beschriebenen werden können oder von unrestringierten, endlichdimensionalen Optimierungsproblemen herkommen, und erklären und bewerten diese; • urteilen über die Stabilität und Effizienz eines numerischen Verfahrens; • setzen mit eigener oder gegebener Software Verfahren um und bewerten deren Ergebnisse kritisch; • erläutern und verwenden ein breites Problem- und Verfahrensspektrum: Differenzenverfahren für Anfangs- und Randwertaufgaben, Finite-Element-Verfahren für 2-Punkt-Randwertaufgaben • übertragen die erlangten Fachkompetenzen auf die Behandlung partieller Differentialgleichungen, Abstiegs- und CG-Verfahren bis zum Barriereverfahren; • sammeln und bewerten relevante Informationen und erkennen Zusammenhänge.
7	**Voraussetzungen für die Teilnahme**	empfohlen: • Analysis • Lineare Algebra

		• Programmierung • Einführung Numerik
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 120 h Eigenstudium: 180 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• P. Deuflhard und F. Bornemann: Numerische Mathematik II; de Gruyter, Berlin 2002 • J. Stoer und R. Bulirsch: Numerische Mathematik II; Springer, Berlin, 2005 • K. Strehmel und R. Weiner: Numerik gewöhnlicher Differentialgleichungen; Teubner, Stuttgart 1995 • A. Quarteroni, R. Sacco und F. Saleri: Numerische Mathematik I, II; Springer, Berlin 2002 • Vorlesungsskriptum auf der Homepage des Bereichs Modellierung, Simulation und Optimierung des Departments Mathematik (laufend aktualisiert)

1	**Modulbezeichnung** 65070	**Einführung in die Darstellungstheorie** Introduction to representation theory	**10 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Peter Fiebig
5	**Inhalt**	• Darstellungen endlicher Gruppen • Module über Ringen • Halbeinfache Ringe • Kategorien und Funktoren • Anwendungen Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • nennen und erläutern die grundlegenden Begriffe der Darstellungstheorie anhand beispielhaft ausgewählter Kapitel und erkennen und erklären deren Zusammenhänge; • ordnen Methoden aus der Algebra in einen übergreifenden Kontext ein und wenden diese an; • analysieren und bewerten algebraische Strukturen und erkennen Zusammenhänge; • klassifizieren und lösen selbstständig algebraische Probleme
7	**Voraussetzungen für die Teilnahme**	empfohlen: Modul Algebra
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung mündlich (20 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) mündlich (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 105 h Eigenstudium: 195 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• C. Meusburger, Vorlesungsskript "Einführung in die Darstellungstheorie" • S. Sternberg, "Group Theory and Physics", CUP 1994

		• M. Artin, "Algebra", Pearson, 2011.

1	**Modulbezeichnung** 65210	**Einführung in die Numerik** Introduction to numerics	**10 ECTS**
2	Lehrveranstaltungen	Vorlesung: Einführung in die Numerik (= Numerische Mathematik) (4 SWS) Übung: Python-Kurs zur Einführung in die Numerik (0 SWS) Übung: Übungen zur Einführung in die Numerik (2 SWS) Übung: Tutorium zur Einführung in die Numerik (1 SWS)	- - - -
3	Lehrende	Dr. Daniel Tenbrinck	

4	**Modulverantwortliche/r**	Prof. Dr. Eberhard Bänsch
5	**Inhalt**	• Direkte Eliminationsverfahren für lineare Gleichungssysteme [Gauß mit Pivotsuche (Erinnerung), Cholesky, LR-Zerlegung für vollbesetzte (Erinnerung) Bandmatrizen] • Linear stationäre iterative Verfahren: Erinnerung und SOR-Verfahren • Verfahren für Eigenwertaufgaben (QR-Verfahren) • Fehleranalyse und Störungsrechnung (Gleitpunktarithmetik, Konditionsanalyse, schlechtgestellte Probleme) • Lineare Ausgleichsrechnung (Orthogonalisierungsverfahren, Numerik der Pseudoinverse) • Iterative Verfahren für nicht-lineare Gleichungssysteme (Fixpunktiteration, Newton-Verfahren, Gauß-Newton) • Interpolation (Polynome, Polynomialsplines, FFT) • Numerische Integration (Newton-Cotes, Gauß, Extrapolation, Adaption) Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • verwenden algorithmische Zugänge für Probleme der linearen Algebra und Analysis und erklären und bewerten diese; • urteilen insbesondere über die Stabilität und Effizienz eines numerischen Verfahrens; • setzen mit eigener oder gegebener Software Verfahren um und bewerten deren Ergebnisse kritisch; • erläutern und verwenden ein breites Problem- und Verfahrensspektrum: (Direkte und) iterative Verfahren für lineare Gleichungssysteme, nicht-lineare Gleichungssysteme, insbesondere Newton-Verfahren, (nicht)lineare Ausgleichsrechnung, Interpolation und Integration, Numerik von Eigenwertaufgaben; • sammeln und bewerten relevante Informationen und erkennen Zusammenhänge

7	**Voraussetzungen für die Teilnahme**	empfohlen: • Module zur Analysis und Linearen Algebra • Kenntnisse in MATLAB sind zwingend. Diese können in einem jeweils vor Semesterbeginn stattfindenden Kurs erworben werden.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 105 h Eigenstudium: 195 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• R. Schaback und H. Wendland: Numerische Mathematik; Springer, Berlin, 2005 • A. Quarteroni, R. Sacco, F. Saleri: Numerische Mathematik I, II; Springer, Berlin, 2002 • P. Deuflhard und A. Hohmann: Numerische Mathematik I; de Gruyter, Berlin 2002 • J. Stoer: Numerische Mathematik I; Springer, Berlin, 2005 • J. Stoer und R. Bulirsch: Numerische Mathematik I; Springer, Berlin, 2005 • Vorlesungsskript auf der Homepage des Bereichs Modellierung, Simulation und Optimierung des Departments Mathematik, ständig neu an die Vorlesung angepasst

1	**Modulbezeichnung** 65580	**Elementare Zahlentheorie** Elementary number theory	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Elementare Zahlentheorie (4 SWS)	-
3	Lehrende	Prof. Dr. Christina Birkenhake	

4	**Modulverantwortliche/r**	Dr. Yasmine Sanderson
5	**Inhalt**	• Anwendung der vollständigen Induktion • Division mit Rest • Untergruppen von Z • ggT und kgV • euklidischer Algorithmus • Teilbarkeitslehre • Begriff der Primzahl und Fundamentalsatz der Arithmetik, Primzahlen und Primzahlprobleme, • Diophantik mit Anwendungen • Prime Restklassengruppe • Dezimalbruch-Entwicklung • Algebraische und transzendente Zahlen Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • nennen und erklären grundlegende Begriffe der elementaren Zahlentheorie; • lösen klassische mathematische Probleme.
7	**Voraussetzungen für die Teilnahme**	empfohlen: • Elemente der Linearen Algebra I und II sowie Elementeder Analysis I und II
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur Übungsleistung
11	**Berechnung der Modulnote**	Klausur (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Vorlesungsskript zu diesem Modul

1	**Modulbezeichnung** 65110	**Funktionalanalysis** Functional analysis	**10 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Günther Grün
5	**Inhalt**	Grundlagen zu folgenden Themen: • Hilbert- und Banach-Räume • Sobolev-Räume • Lineare Operatoren • Lineare Funktionale und der Satz von Hahn-Banach • Prinzip der gleichmäßigen Beschränktheit • Kompakte Operatoren • Lösbarkeit linearer Gleichungen (inklusive Fredholm'sche Alternative) • Spektraltheorie kompakter Operatoren und Anwendungen Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • nennen und erklären die Grundprinzipien der linearen Funktionalanalysis und verwenden diese; • kennen und erklären die Topologien von Hilbert- und Banachräumen, weisen Konvergenz von Folgen in unterschiedlichen Topologien nach (stark, schwach) und zeigen Implikationen aus kompakten Einbettungen auf; • beweisen Aussagen zu Existenz und Eindeutigkeit von Lösungen linearer Operatorgleichungen und zeigen insbesondere die Existenz schwacher Lösungen zu Randwertproblemen bei linearen elliptischen Differentialgleichungen; • treffen Aussagen zur Integrierbarkeit bzw. Glattheit von Sobolev-Funktionen.
7	**Voraussetzungen für die Teilnahme**	empfohlen: Drei der vier Module Lineare Algebra I und II, Analysis I und II müssen bestanden sein.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich (20 Minuten) Übungsleistung
11	**Berechnung der Modulnote**	mündlich (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	nur im Sommersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 210 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Vorlesungsskripte zu diesem Modul • H.W. Alt: Lineare Funktionalanalysis; Springer • D. Werner: Funktionalanalysis; Springer

1	**Modulbezeichnung** 65351	**Funktionentheorie I** Complex analysis I	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Hermann Schulz-Baldes
5	**Inhalt**	Grundlagen zu folgenden Themen: • Holomorphe Abbildungen • Cauchy-Riemann'sche Differentialgleichungen • Wegintegrale und der Cauchy'sche Integralsatz • Satz von Liouville • Laurent-Reihen • Residuenkalkül Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • erklären die Grundprinzipien der Funktionentheorie und wenden diese an; • erkennen die Gemeinsamkeiten und Unterschiede zwischen reell und komplex differenzierbaren Funktionen und erklären diese; • wenden komplex-analytische Methoden zur Lösung von Problemen der reellen Analysis selbständig an.
7	**Voraussetzungen für die Teilnahme**	empfohlen: Analysis I und II
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur mit Übungsleistung
11	**Berechnung der Modulnote**	Klausur mit Übungsleistung (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Freitag, Busam: Funktionentheorie I • Remmert: Funktionentheorie

1	**Modulbezeichnung** 65621	**Geometrie** Geometry	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Friedrich Knop
5	**Inhalt**	Dieses Modul wird mit wechselnden Schwerpunkten angeboten: • Euklidische, hyperbolische, sphärische und projektive Geometrie (Symmetriegruppen geometrischer Strukturen, Invarianten, Geodäten, Dreiecke, Krümmung) • Elementare Differentialgeometrie: Kurventheorie (ebene Kurven, Raumkurven), Flächentheorie (Fundamentalformen, Krümmung, Integration, spezielle Klassen, Riemannsche Metriken) • Algebraische Geometrie: Kommutative Algebra, Nullstellensatz, Affine Varietäten, Projektive Varietäten, Normalisierung, Singularitäten, Algebraische Gruppen Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • wenden Methoden einer der Vertiefungsrichtungen der Geometrie an; • analysieren konkrete Beispiele systematisch und behandeln diese im Rahmen der allgemeinen Theorie.
7	**Voraussetzungen für die Teilnahme**	empfohlen: Die Module der Linearen Algebra, Analysis und Algebra
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur mit Übungsleistung
11	**Berechnung der Modulnote**	Klausur mit Übungsleistung (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Bekanntgabe in der Vorlesung

1	**Modulbezeichnung** 65100	**Gewöhnliche Differentialgleichungen** Ordinary differential equations	**10 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Hermann Schulz-Baldes
5	**Inhalt**	Grundlagen zu folgenden Themen: • Typen von Differentialgleichungen und elementare Lösungsmethoden • Existenz-, Eindeutigkeits- und Stetigkeitssätze für das Anfangswertproblem • Differentialungleichungen (Lemma von Gronwall) • Fortsetzung von Lösungen • lineare und gestörte lineare Systeme • autonome Systeme und Flüsse • Stabilität • Randwertprobleme Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • lösen einfache, insbesondere autonome lineare Differentialgleichungen selbständig • erklären und prüfen qualitative Eigenschaften wie Stabilität • wenden die relevanten Lösungsmethoden selbstständig an • klassifizieren konkrete Probleme und setzen theoretische Modelle zur Behandlung ein • überführen die Prinzipien in allgemeineren oder auch einfacheren Kontext
7	**Voraussetzungen für die Teilnahme**	empfohlen: Analysis 1 und 2
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) Übungsleistung
11	**Berechnung der Modulnote**	Klausur (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 210 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Vorlesungsskripte zu diesem Modul • H. Amann: Gewöhnliche Differentialgleichungen. de Gruyter • V.I. Arnol'd: Gewöhnliche Differentialgleichungen. Springer • H. Heuser: Gewöhnliche Differentialgleichungen. Teubner • W. Walter: Gewöhnliche Differentialgleichungen. Springer

1	**Modulbezeichnung** 48071	**Introduction to Statistics and Statistical Programming** Introduction to statistics and statistical programming	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten. Review session: participation voluntary	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Christophorus Richard
5	**Inhalt**	• Introduction to the statistical software R and elementary programming • Descriptive statistics: visualisation and parameters of categorial and metric data, qq-plot, curve fitting, log- and loglog-plots, robust techniques • Inferential statistics: methods for estimating and testing: parametric tests, selected non-parametric tests, exact and asymptotic confidence regions • Simulation: random numbers, Monte carlo
6	**Lernziele und Kompetenzen**	The students are able to • describe and explain standard techniques in descriptive and inferential statistics. • explain their solution of a non-trivial statistical problem to other people and to discuss alternative solutions within a group. • perform statistical standard analyses within a prescribed time limit on the computer, and to correctly interpret the computer output. • perform elementary statistical simulations. • formulate adequate questions concerning a given data set, suggest correct methods for analysis, and to implement these on the computer.
7	**Voraussetzungen für die Teilnahme**	Stochastische Modellbildung (strongly recommended)
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Lecture notes • Rice: Mathematical Statistics and Data Analysis; Thomson, 2007 • www.cran.r-project.org

1	**Modulbezeichnung** 65979	**Kryptographie I** Cryptography I	**10 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	
5	**Inhalt**	• Einführung in die Kryptographie • Klassische Chiffrierverfahren • Grundeigenschaften der Ringe Z und Z/nZ • Primzahltests • Public-Key-Kryptosysteme RSA • Die Pollard-rho-Methode zur Faktorisierung • Kryptographische Anwendungen diskreter Logarithmen • Kryptographische Hashfunktionen • Digitale Signaturen • Methoden zur Berechnung diskreter Logarithmen • Enigma Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • erklären wichtige kryptographische Verfahren und wenden diese praktisch an • nützen Software wie Maple, Python3 oder Sage zur Ver- und Entschlüsselung sowie zur Kryptoanalyse • erläutern wichtige zahlentheoretische Algorithmen, ihre theoretischen Hintergründe und ihre Funktion bei der Konstruktion von Public-Key-Kryptosystemen
7	**Voraussetzungen für die Teilnahme**	empfohlen: • Grundkenntnisse aus den Modulen Analysis I und Lineare Algebra I
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 210 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Vorlesungsskript zum Modul • J. Buchmann: Einführung in die Kryptographie • J. Hoffstein, J. Pipher, J. H. Silvermann: An Introduction to Mathematical Cryptography

1	**Modulbezeichnung** 65980	**Kryptographie II** Cryptography II	**10 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	
5	**Inhalt**	Die Vorlesung wird mit wechselnden Schwerpunkten angeboten, wobei jeweils ein spezielles zahlentheoretisches Gebiet (wie elliptische Kurven, quadratische Zahlkörper, Gitter) die Grundlage für kryptographische Anwendungen bildet. Die Präsentation des Stoffes erfolgt in Vorlesungsform.
6	**Lernziele und Kompetenzen**	Die Studierenden • erklären fortgeschrittene kryptographische Verfahren und ihre mathematischen Hintergründe • setzen geeignete Software zum praktischen Umgang mit den besprochenen Kryptosystemen ein
7	**Voraussetzungen für die Teilnahme**	empfohlen: • Kryptographie I • Algebra
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 210 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Vorlesungsskript zum Modul

1	**Modulbezeichnung** 65161	**Lineare und Kombinatorische Optimierung** Linear and combinatorial optimisation	**10 ECTS**
2	Lehrveranstaltungen	Übung: Übung zur Linearen und Kombinatorischen Optimierung (2 SWS) Vorlesung: Lineare und Kombinatorische Optimierung (4 SWS)	- -
3	Lehrende	Dr. Dieter Weninger	

4	**Modulverantwortliche/r**	Dr. Dieter Weninger
5	**Inhalt**	Schwerpunkt dieser Vorlesung ist die Theorie und Lösung kombinatorischer und in diesem Kontext linearer Optimierungsprobleme. Wir behandeln klassische Probleme auf Graphen, wie das Kürzeste-Wege-Problem, das Aufspannende-Baum-Problem oder das Max-Flow-Min-Cut-Theorem. Zur Vorlesung gehören auch die Dualität der linearen Optimierung und das Simplexverfahren. Gegenstand der Vorlesung ist zudem die Analyse von Algorithmen und die Vermittlung algorithmischer Grundprinzipien. Neben der vierstündigen Vorlesung werden zweistündige Übungen angeboten. Anhand von Präsenz- und Hausaufgaben werden wesentliche Lerninhalte geübt. Zusätzlich werden Softwareübungen angeboten.
6	**Lernziele und Kompetenzen**	Die Studierenden • erkennen und analysieren selbstständig kombinatorische Optimierungsprobleme; • erläutern algorithmische Grundprinzipien und wenden diese zielorientiert an; • klassifizieren komplexe Verfahren des Lerngebietes; • sammeln und bewerten relevante Informationen und stellen Zusammenhänge her
7	**Voraussetzungen für die Teilnahme**	empfohlen: Lineare Algebra
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 210 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch

16	**Literaturhinweise**	• Vorlesungsskript zu diesem Modul • Schrijver: Combinatorial Optimization Vol. A C; Springer, 2003 • Korte, J. Vygen: Combinatorial Optimization; Springer, 2005

1	**Modulbezeichnung** 65255	**Mathematische Modellierung Praxis** Mathematical modelling practical	**5 ECTS**
2	Lehrveranstaltungen	Projektseminar: MaMoPra (2 SWS)	5 ECTS
3	Lehrende	PD Dr. Florian Frank	

4	**Modulverantwortliche/r**	PD Dr. Serge Kräutle
5	**Inhalt**	
6	**Lernziele und Kompetenzen**	Die Studierenden: • bearbeiten Modellierungsprojekte im Team; • modellieren Alltagsprobleme, lösen sie mit analytischen / numerischen Methoden und diskutieren die Ergebnisse kritisch; • prägen Problemlösungskompetenz aus; • erwerben Schlüsselkompetenzen: prägen durch die Projektarbeit Teammanagement aus, sind durch Berichterstattung in den Projekten zu Vortragspräsentation und wissenschaftlichem Schreiben befähigt.
7	**Voraussetzungen für die Teilnahme**	empfohlen: • Teilnahme am Modul nur in Kombination mit dem Modul Mathematische Modellierung Theorie • Module Analysis und Lineare Algebra oder Module einer zwei-semestrigen Mathematikgrundausbildung für nicht-mathematische Studiengänge, Modul Numerische Mathematik, Modul Gewöhnliche Differentialgleichungen
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminararbeit+Vortrag
11	**Berechnung der Modulnote**	Seminararbeit+Vortrag (0%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Ch. Eck, H. Garcke, P. Knabner: Mathematische Modellierung; Springer-Verlag, 2. Auflage, Berlin, 2011 • F. Hauser, Y. Luchko: Mathematische Modellierung mit MATLAB; Spektrum Akademischer Verlag, 2011 • G. Strang: Introduction to Applied Mathematics; Wellesley-Cambridge Press, Wellesley 1986

1	**Modulbezeichnung** 65254	**Mathematische Modellierung Theorie** Mathematical modelling theory	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Mathematische Modellierung Theorie (2 SWS) Übung: MaMoTheU (2 SWS)	- -
3	Lehrende	PD Dr. Serge Kräutle	

4	**Modulverantwortliche/r**	PD Dr. Serge Kräutle
5	**Inhalt**	• Handwerkszeuge der mathematischen Modellierung: Dimensionsanalyse, asymptotische Entwicklung, Stabilitäts-, Sensitivitätsbetrachtungen, Existenz und Nichtnegativität von Lösungen • Modelle in Form von linearen Gleichungssystemen (elektrische Netzwerke, Stabwerke, Zusammenhang zu Minimierungsaufgaben), nicht-linearen Gleichungssystemen (chemisches Gleichgewicht in reaktiven Mehrspeziessystemen), Anfangswertaufgaben für gewöhnliche Differentialgleichungen (chemische Reaktionen, Populationsmodelle) Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • nennen und erklären die grundlegenden und vertiefenden Begriffe mathematischer Modellierung und verwenden die zugehörigen Prinzipien; • erstellen und bewerten, auf Basis exemplarischer Kenntnisse aus Ingenieur- und Naturwissenschaften, deterministische Modelle in Form von Gleichungssystemen und gewöhnlichen Differentialgleichungen selbstständig; • lösen vorgegebene Aufgaben mit analytischen / numerischen Methoden und diskutieren die Ergebnisse kritisch.
7	**Voraussetzungen für die Teilnahme**	empfohlen: • Teilnahme am Modul nur in Kombination mit dem Modul Mathematische Modellierung Praxis • Module Analysis und Lineare Algebra oder Module einer zwei-semestrigen Mathematikgrundausbildung für nicht-mathematische Studiengänge, Modul Numerische Mathematik, Modul Gewöhnliche Differentialgleichungen empfohlen
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich (15 Minuten)

11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Ch. Eck, H. Garcke, P. Knabner: Mathematische Modellierung; Springer-Verlag, 2. Auflage, Berlin, 2011 • F. Hauser, Y. Luchko: Mathematische Modellierung mit MATLAB; Spektrum Akademischer Verlag, 2011 • G. Strang: Introduction to Applied Mathematics; Wellesley-Cambridge Press, Wellesley, 1986

1	**Modulbezeichnung** 65150	**Nichtlineare Optimierung** Nonlinear optimisation	**10 ECTS**
2	Lehrveranstaltungen	Übung: Übungen zu Nichtlinearer Optimierung (2 SWS) Vorlesung: Nichtlineare Optimierung (4 SWS)	- -
3	Lehrende	Dr. Lukas Pflug	

4	**Modulverantwortliche/r**	Prof. Dr. Wolfgang Achtziger
5	**Inhalt**	• Unrestringierte Probleme der Nichtlinearen Optimierung (Optimalitätsbedingungen, Abstiegsverfahren, Verfahren der konjugierten Richtungen, Variable-Metrik-Methoden und Quasi-Newton-Methoden) • Restringierte Probleme der Nichtlinearen Optimierung (Optimalitätsbedingungen) Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • nennen und erklären Grundbegriffe der Nichtlinearen Optimierung; • modellieren und lösen praxisrelevante Probleme mit Hilfe der erlernten Verfahren; • sammeln und bewerten relevante Informationen und stellen Zusammenhänge her.
7	**Voraussetzungen für die Teilnahme**	empfohlen: Abschluss der Module Analysis I, Analysis II, Lineare Algebra I, Lineare Algebra II und Numerische Mathematik.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 210 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Geiger, Ch. Kanzow: Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben; Springer, 1999 • Geiger, Ch. Kanzow: Theorie und Numerik restringierter Optimierungsaufgaben; Springer, 2002 • W. Alt: Nichtlineare Optimierung; Vieweg, 2002

		• F. Jarre und J. Stoer: Optimierung; Springer, 2004 • M.S. Bazaraa, H.D. Sherali, C.M. Shetty: Nonlinear Programming Theory and Algorithms; Wiley, New York, 1993

1	**Modulbezeichnung** 407487	**Numerical Aspects of Linear and Integer Programming** Numerical aspects of linear and integer programming	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Frauke Liers-Bergmann
5	**Inhalt**	• Revidiertes Simplexverfahren (mit Schranken) • Phase I des Verfahrens • Duales Simplexverfahren • LP Presolve/Postsolve • Skalierung • MIP Solution Techniques Die Präsentation des Stoffes erfolgt in Vorlesungsform. In den Übungen werden die Studierenden von einem Übungsgruppenleiter betreut.
6	**Lernziele und Kompetenzen**	Die Studierenden erklären und verwenden im Rahmen der Vorlesung Methoden und numerische Verfahren, die zur Lösung von Linearen und Gemischt-ganzzahligen Programmen in der Praxis Anwendung finden.
7	**Voraussetzungen für die Teilnahme**	empfohlen: Lineare Algebra, Lineare und Kombinatorische Optimierung
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 36 h Eigenstudium: 114 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• V. Chvátal: Linear Programming, W. H. Freeman and Company, New York, 1983 • L.A. Wolsey: Integer Programming, John Wiley and Sons, Inc., 1998

1	**Modulbezeichnung** 64620	**Numerik I für Ingenieure** Numerics for engineers I	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Numerik I für Ingenieure (2 SWS) Praktikum: Übungen Numerik 1 für Ingenieure (2 SWS)	- -
3	Lehrende	PD Dr. Nicolas Neuß	

4	**Modulverantwortliche/r**	Dr. Michael Fried apl.Prof.Dr. Wilhelm Merz
5	**Inhalt**	• Elementare Numerik: Direkte und iterative Lösungsverfahren bei linearen Gleichungssystemen,Interpolation mit Newton-Polynomen und Splines, Quadratur mit Newton-Côtes-Formeln, Extrapolation nach Romberg • Numerik gewöhnlicher Differentialgleichungen: Verschiedene Runge-Kutta Methoden als Einschrittverfahren, Konsistenz, Stabilität- und Konvergenzaussage, Mehrschrittverfahren
6	**Lernziele und Kompetenzen**	Die Studierenden lernen • verschiedene numerische Methoden zur Lösung linearer Gleichungssysteme • verschiedene Methoden zu beurteilen • Interpolationstechniken und Güte der Approximation • grundlegende Quadraturverfahren und die Beurteilung solcher • grundlegende Diskretisierungsmethoden bei gewöhnlichen Differentialgleichungen • Beurteilung dieser Methoden und Verfahren • algorithmische Umsetzung o.g. Verfahren als Grundlage für Computer-Codes
7	**Voraussetzungen für die Teilnahme**	Kurse Mathematik für Ingenieure I, II und III
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	• Skripte des Dozenten • H.-R. Schwarz, N. Köckler: |Numerische Mathematik|, Teubner

1	**Modulbezeichnung** 64631	**Numerik II für Ingenieure** Numerics for engineers II	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	apl.Prof.Dr. Wilhelm Merz
5	**Inhalt**	*Numerik partieller Differentialgleichungen* Finite Differenzenmethode, Stabilität, Konsistenz, Konvergenz, Einführung finite Elementmethode bei elliptischen Problemen, Fehlerschätzer
6	**Lernziele und Kompetenzen**	Die Studierenden • erklären verschiedene Diskretisierungsmethoden • beurteilen diese Diskretisierungsmethoden • leiten Finite Elemente Diskretisierungen elliptischer Probleme her • folgern Aussagen anhand grundlegender Beweistechniken aus oben genannten Bereichen • konstruieren Algorithmen zu Finite Elemente Diskretisierungen • erklären Fehlerschätzer
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Skripte des Dozenten H. Jung, M. Langer, Methode der Finiten Elemente, Teubner P. Knabner, L. Angermann, Numerik partieller Differentialgleichungen, Springer

1	**Modulbezeichnung** 65991	**Operations Research 2** Operations research 2	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übung zur Linearen und Kombinatorischen Optimierung (2 SWS)	-
3	Lehrende	Dr. Dieter Weninger	

4	**Modulverantwortliche/r**	Prof. Dr. Alexander Martin
5	**Inhalt**	Schwerpunkt dieser Vorlesung ist die Theorie und Lösung kombinatorischer und in diesem Kontext linearer Optimierungsprobleme. Wir behandeln klassische Probleme auf Graphen, wie das Kürzeste-Wege-Problem, das Aufspannende-Baum-Problem oder das Max-Flow-Min-Cut-Theorem. Zum Vorlesungsumfang gehört auch das Simplexverfahren für lineare Programme und das Studium algorithmischer Grundprinzipien wie Sortieren, Greedy, Tiefen- und Breitensuche sowie Heuristiken.
6	**Lernziele und Kompetenzen**	Die Studierenden • erkennen und analysieren selbstständig kombinatorische Optimierungsprobleme; • erläutern algorithmische Grundprinzipien und wenden diese zielorientiert an; • klassifizieren komplexe Verfahren des Lerngebietes; • sammeln und bewerten relevante Informationen und stellen Zusammenhänge her
7	**Voraussetzungen für die Teilnahme**	Mathematik Pflichtkurse aus dem Bachelorprogramm
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 45 h Eigenstudium: 105 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Vorlesungsskript zu diesem Modul • Schrijver: Combinatorial Optimization Vol. A C; Springer, 2003 • Korte, J. Vygen: Combinatorial Optimization; Springer, 2005

1	**Modulbezeichnung** 44050	**Optimierung für Ingenieure** Optimisation for engineers	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Dr. Johannes Hild
5	**Inhalt**	**Introduction to continuous optimization problems and methods with and without constraints** • Classification of problem types • Optimality conditions and termination criterions • Descent directions and line search methods • Convergence analysis **Unconstrained optimization** • Steepest descent and conjugate gradient • Newton-type methods • Nonlinear Least Squares **Constrained optimization** • Projection methods • Trust Region • Barrier and penalty methods • Interior point methods **Noisy Functions** • Simplex Gradient • Implicit Filtering
6	**Lernziele und Kompetenzen**	**Competences** **Know** • Students list requirements, strengths and weaknesses of common optimization methods. • Students recognize crucial components in existence and convergence proofs in the context of minimizing sequences. **Understand** • Students explain the different components of optimization methods. • Students describe the relationship between requirements and conclusions of existence and convergence theorems in the context of minimizing sequences. **Apply** • Students check feasibility, well-posedness and constraint qualifications of optimization problems. • Students formulate and solve optimality conditions analytically. • Students apply optimization algorithms to optimization problems. **Analyse** • Students analyse uncommon optimization approaches and extract their requirements, strengths and weaknesses. **Evaluate**

		• Students evaluate the class and structure of unsolved optimization problems. • Students choose suitable algorithmic approaches for unsolved optimization problems. **Create** • Students formulate optimization problems using mathematical methods and structures. • Students modify and combine common optimization routines to create project-specific algorithms for unsolved optimization problems.
7	**Voraussetzungen für die Teilnahme**	Requires contents of the lecture Mathematics for Engineers I, II and III. Especially: • Linear algebra • Analysis of real valued functions • Differential and integral calculus in multi dimensional spaces Requires successful participation in the weekly e-learning assessments.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur 5 ECTS: Written exam open book online based on the content of the lecture.
11	**Berechnung der Modulnote**	Klausur (100%) The grade of the module equals the grade of the written exam.
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 45 h Eigenstudium: 105 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Nocedal, Jorge and Wright, Stephen J.: Numerical Optimization. Springer Serie in Operations Research, 2006. Kelley, C. T.: Iterative Methods for Optimization. Frontiers in Applied Mathematics 18, SIAM Philadelphia 1999; Polak, E.: Optimization. Algorithms and Consistent Approximations.Applied Mathematical Sciences, Volume 124, Springer-Verlag New York, 1997. Jarre, F.:Optimierung, Springer 2003;

1	**Modulbezeichnung** 44060	**Optimierung für Ingenieure mit Praktikum** Optimisation for engineers (with laboratory)	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Dr. Johannes Hild
5	**Inhalt**	**Introduction to continuous optimization problems and methods with and without constraints** • Classification of problem types • Optimality conditions and termination criterions • Descent directions and line search methods • Convergence analysis **Unconstrained optimization** • Steepest descent and conjugate gradient • Newton-type methods • Nonlinear Least Squares **Constrained optimization** • Projection methods • Trust Region • Barrier and penalty methods • Interior point methods **Noisy Functions** • Simplex Gradient • Implicit Filtering **Programming Laboratory** • Implementation of optimization algorithms • Algorithmic optimization of test problems • Solving a benchmark problem
6	**Lernziele und Kompetenzen**	**Competences** **Know** • Students list requirements, strengths and weaknesses of common optimization methods. • Students recognize crucial components in existence and convergence proofs in the context of minimizing sequences. • Students identify optimization routines written in a programming language. **Understand** • Students explain the different components of optimization methods. • Students describe the relationship between requirements and conclusions of existence and convergence theorems in the context of minimizing sequences. **Apply** • Students check feasibility, well-posedness and constraint qualifications of optimization problems. • Students formulate and solve optimality conditions analytically.

		• Students apply optimization algorithms to optimization problems. **Analyse** • Students analyse uncommon optimization approaches and extract their requirements, strengths and weaknesses. • Students observe the behavior of common optimization algorithms applied to numerical test problems. **Evaluate** • Students evaluate the class and structure of unsolved optimization problems. • Students choose suitable algorithmic approaches for unsolved optimization problems. **Create** • Students formulate optimization problems using mathematical methods and structures. • Students modify and combine common optimization routines to create project-specific algorithms for unsolved optimization problems. • Students plan and implement optimization algorithms in a programming language.
7	**Voraussetzungen für die Teilnahme**	Requires contents of the lecture Mathematics for Engineers I, II and III. Especially: • Linear algebra • Analysis of real valued functions • Differential and integral calculus in multi dimensional spaces Requires successful participation in the weekly e-learning assessments and lab programming works. Requires basic knowledge in the implementation of algorithms and data structures in a development environment.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur Übungsleistung 5 ECTS: Written exam open book online based on the content of the lecture. 2.5 ECTS: Completing the programming homework assignments of the laboratory within the specific deadlines. Both parts can be done independently from each other.
11	**Berechnung der Modulnote**	Klausur (100%) Übungsleistung (0%) The grade of the module equals the grade of the written exam.

		The laboratory is only pass or fail and does not influence the grade of the module.
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 165 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Nocedal, Jorge and Wright, Stephen J.: Numerical Optimization. Springer Serie in Operations Research, 2006. Kelley, C. T.: Iterative Methods for Optimization. Frontiers in Applied Mathematics 18, SIAM Philadelphia 1999; Polak, E.: Optimization. Algorithms and Consistent Approximations.Applied Mathematical Sciences, Volume 124, Springer-Verlag New York, 1997. Jarre, F.:Optimierung, Springer 2003;

1	**Modulbezeichnung** 65923	**Optimization in industry and economy**	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Optimization in Industry and Economy (2 SWS) Übung: Übung zu Optimization in Industry and Economy (1 SWS)	5 ECTS -
3	Lehrende	Daniela Bernhard Prof. Dr. Sascha Kurz Jorge Weston Fernández	

4	**Modulverantwortliche/r**	Prof. Dr. Frauke Liers-Bergmann
5	**Inhalt**	This course focuses on modeling and solving real-world optimization problems occurring in industry and economics. Advantages and disadvantages of different modeling techniques will be outlined. In order to achieve efficient solution approaches, different reformulations and their numerical results will be discussed. Students will learn how to present optimization results properly as well as how to interpret and evaluate these results for practical applications. The latter may include but is not limited to the optimization of transport networks (gas, water, energy), air traffic management and mathematical modeling/optimization of market mechanisms in the energy sector.
6	**Lernziele und Kompetenzen**	Students • model complex real-world optimization problems with respect to efficient • solvability, • classify the models and use appropriate solution strategies, • evaluate the achieved computational results.
7	**Voraussetzungen für die Teilnahme**	Recommended: Modul LKOpt: Linear and combinatorial optimization
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 45 h Eigenstudium: 105 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Lecture notes (will be published on StudOn at the beginning of the semester) • Up-to-date research literature (will be published on StudOn at the beginning of the semester)

1	**Modulbezeichnung** 65175	**Robuste Optimierung 1** Robust optimization	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Frauke Liers-Bergmann
5	**Inhalt**	Oft sind die Eingabedaten eines mathematischen Optimierungsproblems in der Praxis nicht exakt bekannt. In der robusten Optimierung werden deswegen möglichst gute Lösungen bestimmt, die für alle innerhalb gewisser Toleranzen liegenden Eingabedaten, zulässig sind. Die Vorlesung behandelt die Theorie und Modellierung robuster Optimierungsprobleme, insbesondere die robuste lineare und robuste kombinatorische Optimierung. Darüber hinaus werden anhand von Anwendungsbeispielen aktuelle Konzepte wie z.B. die wiederherstellbare Robustheit gelehrt. Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • erkennen selbstständig Optimierungsprobleme unter Unsicherheit, modellieren die zugehörigen robustifizierten Optimierungsprobleme geeignet und analysieren diese; • nutzen die passenden Lösungsverfahren und bewerten die erzielten Ergebnisse.
7	**Voraussetzungen für die Teilnahme**	empfohlen: Lineare Algebra Vorteilhaft ist das Modul Lineare und Kombinatorische Optimierung.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Vorlesungsskript zu diesem Modul • Ben-Tal, El Ghaoui, Nemirovski: Robust Optimization; Princeton University Press

1	**Modulbezeichnung** 65918	**Robust optimization II**	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Frauke Liers-Bergmann
5	**Inhalt**	• In practice, provided data for mathematical optimization problems is often not fully known. Robust optimization aims at finding the best solution which is feasible for input data varying within certain tolerances. The lecture covers advanced methods of robust optimization in theory and modeling. In particular, robust network flows, robust integer optimization and robust approximation are included. Further, state-of-the-art concepts, e.g., "light robustness" or "adjustable robustness" will be discussed by means of real-world applications.
6	**Lernziele und Kompetenzen**	Students • will be able to identify complex optimization problems under uncertainties as well as suitably model and analyze the corresponding robust optimization problem with the help of advanced techniques of robust optimization, • learn the handling of appropriate solving techniques and how to analyze the corresponding results.
7	**Voraussetzungen für die Teilnahme**	• Recommended: Robust Optimization I
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 45 h Eigenstudium: 105 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Lecture notes, will be published on StudOn at the beginning of the semester.

1	**Modulbezeichnung** 65080	**Topologie** Topology	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Karl-Hermann Neeb
5	**Inhalt**	• Stetige Funktionen, Zusammenhang, Trennungsaxiome • Erzeugung von Topologien (initiale, finale, Quotienten etc.) • Konvergenz in topologischen Räumen (Filter, Netze) • Kompaktheit (Satz von Tychonov, kompakte metrische Räume, lokalkompakte Räume) • Anwendung auf Funktionenräume (Satz von Stone-Weierstraß, Satz von Ascoli) Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch Übungen
6	**Lernziele und Kompetenzen**	Die Studierenden • wenden die Methoden der allgemeinen Topologie, die in den Grundvorlesungen nur am Rande vorkommen, an; • ordnen die topologischen Grundbegriffe in einen größeren Kontext ein; erklären und verwenden wichtige Resultate, die in vielen Bereichen der Mathematik zum Handwerkzeug gehören
7	**Voraussetzungen für die Teilnahme**	empfohlen: Grundkenntnisse aus den Modulen Analysis I und II
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Mathematik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Vorlesungsskript zu diesem Modul

Nebenfach Artificial Intelligence in Biomedical Engineering

1	**Modulbezeichnung** 96837	**A look inside the human body - gait analysis and simulation**	**2,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: A look inside the human body - gait analysis and simulation (2 SWS)	2,5 ECTS
3	Lehrende	Prof. Dr. Anne Koelewijn	

4	**Modulverantwortliche/r**	Prof. Dr. Anne Koelewijn
5	**Inhalt**	The aim of this lecture is to teach methods of gait analysis and simulation. Gait analysis experiments will be covered, as well as more modern approaches to gather walking data. Techniques to process gait analysis experiments are discussed, as well as dynamic models that can be used to create gait simulations. This lecture addresses the following topics: • Measurement systems for gait analysis • Methods to calculate joint kinetics and kinematics from experimental data • Muscle biology, specific to force generation, and modelling of muscles • Methods to calculate muscle activation from experimental data • Energetics of walking • Multibody dynamics • Creating simulations of gait
6	**Lernziele und Kompetenzen**	Learning objectives: • Be familiar with the existing measurement options for gait analysis • Know state-of-the art techniques to process gait analysis experiments • Select an appropriate processing technique for a specific experiment • Understand how gait could be simulated and where these simulations could be applied • Know the function of the different components of the human body that are involved in locomotion
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Artificial Intelligence in Biomedical Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 45 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Winter, David A. Biomechanics and motor control of human movement. John Wiley & Sons, 2009. • Kelly, Matthew. "An introduction to trajectory optimization: How to do your own direct collocation." SIAM Review 59.4 (2017): 849-904.

1	**Modulbezeichnung** 47666	**Bewegungsanalyse und biomechanische Grenzgebiete** Motion analysis and biomechanical boundary areas	**2,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Bewegungsanalyse und biomechanische Grenzgebiete (2 SWS)	2,5 ECTS
3	Lehrende	Prof. Dr. Anne Koelewijn	

4	**Modulverantwortliche/r**	Prof. Dr. Anne Koelewijn Prof. Dr.-Ing. Sigrid Leyendecker
5	**Inhalt**	Anatomie des menschlichen Bewegungsapparates |Muskeln, Sehnen, Bänder, Knochen, Knorpel| Gelenkmechanik Kinematik |Bewegungsanalyse und Motion-Capturing-Systeme| Kinetik |Kraft- & Druckmessplatten, Bodenreaktionskräfte| Elektromyographie 3D-Modellierung in der Biomechanik |Segmentierung, 3D-Modelle| Simulation |FEM, MKS|
6	**Lernziele und Kompetenzen**	Nach dem erfolgreichen Absolvieren des Moduls verfügen die Studierenden über die folgenden fachlichen, methodischen und persönlichen Kompetenzen: Die Studierenden verstehen die in der Biomechanik verwendete Technik und angewandte Methoden. Die Studierenden bestimmen die anatomischen Strukturen, die den aktiven bzw. passiven Bewegungsapparat aufbauen und somit Kraftentwicklung und Bewegungen ermöglichen. Die Studierenden sind in der Lage die in der technischen Mechanik erlernten Größen wie Kräfte und Momente auf anatomische Strukturen anzuwenden. Die Studierenden können die Funktion der im Rahmen der Kinematik verwendeten Systeme zum Motion-Capturing beschreiben und gegenüberstellen. Sie sind in der Lage, die in der Kinetik verwendeten Messsysteme wie Kraft- und Druckmessplatten in Aufbau und Funktion zu unterscheiden. Sie können die gemessenen Bodenreaktionskräfte und Kraft-Zeit-Verläufe interpretieren und in Zusammenhang mit Bewegungen und Kraftübertragung setzen. Die Studierenden sind in der Lage ein Vorgehen zur Messung von Muskelaktivitäten bei einer spezifischen Bewegung durch Elektromyographie zu entwerfen. Sie beschreiben die Funktion von EMG-Sensoren, unterschiedliche Filtertechniken, Arten der Ableitung wie auch Einflussfaktoren und erläutern diese. Die Studierenden beschreiben die Vorzüge der 3D-Modellierung im biomechanischen und orthopädischen Umfeld und können die unterschiedlichen Arten der Segmentierung gegenüberstellen.

		Die Studierenden sind in der Lage, Simulation im Maschinenbau und in der Biomechanik gegenüberzustellen und die Unterschiede zu konkretisieren. Sie beschreiben den grundsätzlichen Aufbau von Finiter-Elemente-Analyse (FEM) und Mehrkörpersimulation (MKS) und begründen die Funktion in biomechanischem Kontext.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Artificial Intelligence in Biomedical Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	elektronische Prüfung
11	**Berechnung der Modulnote**	elektronische Prüfung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 75 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Relevante Literatur ist im online-Kurs zu den jeweiligen Kapiteln angegeben.

1	**Modulbezeichnung** 44200	**Computational Neurotechnology / Numerische Neurotechnologie** Computational neurotechnology / Numerical neurotechnology	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Tobias Reichenbach
5	**Inhalt**	Foundations of Computational Neuroscience and the processing of neural signals. Applications in the areas of artificial neural networks, Brain-Machine-Interfaces (BCIs) and neural prosthesis.
6	**Lernziele und Kompetenzen**	• Can understand the principles of the analysis of neural signals • Can apply information theory for the description of neural activity • Can perform simulations of the dynamics of single neurons as well as of neural networks • Can evaluate different approaches to construct Brain-Machine-Interfaces (BCIs) • Can explain concepts for the design of neural prosthesis
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Artificial Intelligence in Biomedical Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 56 h Eigenstudium: 94 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Dayan, Peter, and Laurence F. Abbott. Theoretical neuroscience: computational and mathematical modeling of neural systems. Computational Neuroscience Series, 2001. Gerstner, Wulfram, et al. Neuronal dynamics: From single neurons to networks and models of cognition. Cambridge University Press, 2014. Oweiss, Karim G., ed. Statistical signal processing for neuroscience and neurotechnology. Academic Press, 2010.

		Maurits, Natasha. From neurology to methodology and back: an introduction to clinical neuroengineering. Springer Science & Business Media, 2011. Clément, Claude. Brain-Computer Interface Technologies. Springer International Publishing, 2019. DiLorenzo, Daniel J., and Joseph D. Bronzino, eds. Neuroengineering. CRC Press, 2007.

1	**Modulbezeichnung** 92358	**Inertial Sensor Fusion** Inertial sensor fusion	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Inertial Sensor Fusion (4 SWS)	5 ECTS
3	Lehrende	Simon Bachhuber Dr. Ive Weygers	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Thomas Seel
5	**Inhalt**	This module is concerned with inertial sensor technologies and sensor fusion methods for motion tracking of aerial/ground/water vehicles, robotic systems and human body segments. Participants will become familiar with the design and application of methods and algorithms for sensor fusion and analysis of inertial measurement data. This includes methods to estimate the orientation and position of moving objects in three-dimensional space as well as methods for calculating joint angles or segmenting human motion. Since most of the considered applications are feedback-controlled systems, the course focuses on real-time-capable algorithms. The methods will be applied to application data during designated computer exercises that are integrated into the course. Topics include, but are not limited to: • Basic principles of gyroscopes, accelerometers and magnetometers • Error characteristics of MEMS-based inertial measurement units • Application: Gait phase detection by foot-worn inertial sensors • Quaternions and other representations of 3D rotations • Orientation estimation from inertial measurement data • Application: Position tracking/retrieval of an unmanned aerial vehicle • Joint angle estimation from inertial measurement data • Application: Real-time motion tracking of a robotic actuator • Kalman filtering methods for linear and nonlinear systems • Probabilistic sensor fusion and Bayesian state estimation • Identification of kinematic parameters from inertial measurement data • Application: Human body motion tracking by wearable inertial sensors
6	**Lernziele und Kompetenzen**	• Students are able to employ inertial sensor technologies and sensor fusion methods for applications in research and industry. • They are capable of understanding and handling the complexity of inertial sensor data and have command of a versatile set of methods for real-time processing of inertial measurements. • They are able toi track the orientation and position of an unmanned aerial vehicle.

		• They are able to track the motion of multi-link kinematic chains, e.g. robotic actuators or human limbs, in three dimensional space.
7	**Voraussetzungen für die Teilnahme**	Participants should be familiar with fundamentals of linear algebra. It is advantageous but not required to be have some prior knowledge on linear dynamic systems or basic probability theory.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Artificial Intelligence in Biomedical Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Woodman, O.J. An Introduction to Inertial Navigation; University of Cambridge, Computer Laboratory: Cambridge, UK, 2007. • T. Seel, M. Kok, R. McGinnis, "Inertial SensorsApplications and Challenges in a Nutshell", Sensors 2020, 20, 6221. • M. Kok, J. D. Hol, and T. B. Schön, "An optimization-based approach to human body motion capture using inertial sensors, IFAC Proceedings Volumes, vol. 47, no. 3, pp. 7985, Jan. 2014. • B. Taetz, G. Bleser, and M. Miezal, "Towards self-calibrating inertial body motion capture, in 2016 19th International Conference on Information Fusion (FUSION), Jul. 2016, pp. 17511759. • D. Lehmann, D. Laidig, and T. Seel, "Magnetometer-free motion tracking of one-dimensional joints by exploiting kinematic constraints, Proceedings on Automation in Medical Engineering, vol. 1, no. 1, pp. 027027, 2020. • D. Laidig, D. Lehmann, M.-A. Bégin, and T. Seel, "Magnetometer-free realtime inertial motion tracking by exploitation of kinematic constraints in 2-dof joints, 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 12331238, 2019. • M. Caruso, A.M. Sabatini, D. Laidig, T. Seel, M. Knaflitz, U. DellaCroce, A. Cereatti. Analysis of the Accuracy of Ten Algorithms for Orientation Estimation Using Inertial and

		Magnetic Sensing under Optimal Conditions: One Size Does Not Fit All. Sensors, 21 (7):2543, 2021. • E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation, in Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373), Oct 2000, pp. 153158. • J. Steinbring and U. D. Hanebeck, "S2kf: The smart sampling kalman filter, in Proceedings of the 16th International Conference on Information Fusion, 2013, pp. 20892096. • A. Solin, S. Särkkä, J. Kannala, and E. Rahtu, "Terrain navigation in the magnetic landscape: Particle filtering for indoor positioning, 05 2016, pp. 19.

1	**Modulbezeichnung** 44157	**Interfacing the Neuromuscular system: Applications for Human/Machine Interfaces and Neurophysiology** Interfacing the neuromuscular system: Applications for Human/machine interfaces and neurophysiology	**5 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Prof. Dr. Alessandro Del Vecchio
5	**Inhalt**	Module: Principles of Neural control of movement and neuroengineering How the central nervous system controls muscle forces; Neurons, upper and lower motoneurons, Cortical and brainstem function, Interneurons and Motor Units. Neuroengineering applications for studying the neural control of movement; invasive and non-invasive recordings, electrical stimulation of the nervous system. Module: Electrophysiology Generation of an action potential; HodgkinHuxley model, difference between intracellular and extracellular action potential, sparsity of the action potential in a matrix of electrodes. Recording electrophysiological data in humans; examples of EMG and EEG recordings. Module: Applications to Human/Machine Interfaces Biosignal processing; data with high temporal resolution, identification of individual neurons, associations between neuronal discharge times and behaviour; control of prosthetic devices from EMG signals in amputees and neurodegenerative and neurotraumatic diseases. Module: Applications to Neurophysiology Neuronal encoding of behaviour; motor unit physiology in humans; motoneuron properties, longitudinal assessment of neuronal function. Module: MATLAB / Python practical coursework Extraction of neural information from electrophysiological signals; associations of information between electrophysiological signals and behavioural data; Experiment in humans.
6	**Lernziele und Kompetenzen**	The students will acquire in-depth skills in the acquisition, analysis, and interpretation of electrophysiological data with a specific focus on human recordings in health and pathological conditions (e.g., spinal cord injury, stroke, and Parkinsons disease). The goal of this course is to teach the current methods in man/machine interfaces and neurophysiological applications. The course will provide information on the neural circuitries that determine coordinated movement. The specific focus is on the motor system that regulates skilled motor behaviour. We will study the physiological pathways of the motor system and the effect of neurodegenerative diseases that affect this system. Ultimately, this course will give students a robust overview of how to use electrophysiology in order to assist individuals with neural impairments.
7	**Voraussetzungen für die Teilnahme**	No compulsory prerequisites.

		Recommended: Basic biology and neurophysiology, Computer programming (Matlab and/or Python), Biosignal processing.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Artificial Intelligence in Biomedical Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur Written examination.
11	**Berechnung der Modulnote**	Klausur (100%) Oral examination 100%
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	
16	**Literaturhinweise**	• Principles of Neuroscience from Eric R. Kandel, MD • Motor unit from Heckman and Enoka, DOI: 10.1002/cphy.c100087 • Surface Electromyography, Physiology, Engineering, and Applications Edited by Roberto Merletti and Dario Farina • Neural Engineering, Edited by Bin He • Tutorial: Analysis of motor unit discharge characteristics from high-density surface EMG signals, Del Vecchio et al. https://doi.org/10.1016/j.jelekin.2020.102426 • Restoring sensorimotor function through intracortical interfaces: progress and looming challenges, Bensmaia and Miller https://www.nature.com/articles/nrn3724

1	**Modulbezeichnung** 47657	**Legged Locomotion of Robots + Laborprojekt (LLR-L)** Legged locomotion of robots + laboratory project (LLR-L)	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Anne Koelewijn
5	**Inhalt**	Legged robotics help researchers understand human and animal locomotion. Furthermore, legged robots have many different applications, for example to aid in dangerous environments and in rehabilitation. Active prosthetics and exoskeletons improve gait of people with a disability, like a spinal cord injury or an amputation. The goal of this seminar is to become familiar with different algorithms and analysis methods that are used for legged robotics. Important concepts here are the energetics and the stability. Robots should be energy efficient, in the case of an exoskeleton to not lose battery power for a day. Obviously, stability is important to avoid falls. Each student will perform a literature review of a specific concept related to robot locomotion. The concepts can be chosen from a list, or the student can propose their own topic. In addition, students will do a lab project. This will require the student to implement the chosen concept in simulation or in practice.

6	**Lernziele und Kompetenzen**	Fachkompetenz
		Wissen
		•
		  ° The students understand the theoretical background of concepts of robot locomotion.
		  ° The students are able to differentiate between different concepts of robot locomotion.
		  ° The students are able to understand the stability and energetics in robot locomotion.
		  ° The students are able to transfer their knowledge about robot locomotion to new use cases. Analysieren The students are able to analyse and discuss new ideas and research potentials for robot locomotion.

7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Artificial Intelligence in Biomedical Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung

11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 47656	**Legged Locomotion of Robots (LLR)** Legged locomotion of robots (LLR)	**2,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Anne Koelewijn
5	**Inhalt**	Legged robotics help researchers understand human and animal locomotion. Furthermore, legged robots have many different applications, for example to aid in dangerous environments and in rehabilitation. Active prosthetics and exoskeletons improve gait of people with a disability, like a spinal cord injury or an amputation. The goal of this seminar is to become familiar with different algorithms and analysis methods that are used for legged robotics. Important concepts here are the energetics and the stability. Robots should be energy efficient, in the case of an exoskeleton to not lose battery power for a day. Obviously, stability is important to avoid falls. Each student will perform a literature review of a specific concept related to robot locomotion. The concepts can be chosen from a list, or the student can propose their own topic. Students can choose to perform an extra assignment to receive an additional 2.5 ECTS. The assignment will require the student to implement the chosen concept in simulation or in practice.
6	**Lernziele und Kompetenzen**	• Familiarize with different concepts that are used in control and analysis of robot locomotion • Understand the theoretical background of concepts of robot locomotion • Differentiate between different types of robots • Understand the stability and energetics in robot locomotion
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Artificial Intelligence in Biomedical Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 45 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch

16	**Literaturhinweise**	

1	**Modulbezeichnung** 47687	**Research Project on Intelligent Sensorimotor Systems** Research project on intelligent sensorimotor systems	**10 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Simon Bachhuber Prof. Dr.-Ing. Thomas Seel
5	**Inhalt**	Participants will work on individual research projects that are related to our current research in the fields of Modeling and Simulation of Dynamical Systems, Sensor Fusion, Hybrid AI, Dynamic Inference and Learning, Combining Systems and Control with Machine Learning for Prevention, Diagnosis and Treatment Solutions. Unlike a course with fixed contents, the project topics are defined individually based on current research questions and personal preferences and prerequisites. This research project is ideal for for a range of degree programs with 10 ECTS projects. However, most projects can also be offered as 5 ECTS medical engineering internship/praktikum. All interested students are kindly asked to contact us by email.
6	**Lernziele und Kompetenzen**	Participants will • systematically analyze a complex research and create solution approaches • devise AI methods and develop algorithms that implement these methods • generate and analyze high-dimensional data set by simulations or experiments • create a versatile and well-documented solution tool chain • evaluate research results to derive precise conclusions • present their method, results and findings
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Artificial Intelligence in Biomedical Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	in jedem Semester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

Nebenfach Medizinische Informatik

1	**Modulbezeichnung** 22101	**Biometrie und Epidemiologie** Biometrics and epidemiology	**5 ECTS**
2	Lehrveranstaltungen	Grundkurs: Datenanalyse in Biometrie und Epidemiologie (BA Mol Medizin) (2 SWS) Vorlesung: Biometrie und Epidemiologie (BA Molekular-Medizin) (1 SWS) Übung: Biometrie und Epidemiologie (BA Molekular-Medizin) (1 SWS)	- 5 ECTS 5 ECTS
3	Lehrende	Prof. Dr. Olaf Gefeller Dr. Sebastian Meyer Prof. Dr. Annette Pfahlberg PD Dr. Benjamin Hofner	

4	**Modulverantwortliche/r**	Prof. Dr. Olaf Gefeller
5	**Inhalt**	• Studienplanung beobachtender und experimentell intervenierender Studien in der Medizin • Epidemiologische Studiendesigns und Maßzahlen • Deskriptive Datenbeschreibung • Grundbegriffe der Wahrscheinlichkeitsrechnung • Korrelations- und Regressionsanalyse • tatistische Methoden zur Evaluation diagnostischer Verfahren • Statistische Schätz- und Testverfahren • Überblick über gebräuchliche statistische Tests • Überlebenszeitanalyse • Einführung in eine statistische Programmiersprache • Umsetzung methodischer Techniken in die praktische Datenanalyse in einer statistischen Programmierumgebung • Projektarbeit Datenanalyse
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben einen Überblick über epidemiologische und biometrische Herangehensweisen und Argumentationen, • erlernen ein breites Spektrum methodischer Techniken der angewandten Statistik zur Analyse der Beziehung zwischen Risikofaktoren und Krankheiten sowie zur Effektivität therapeutischer Interventionsmaßnahmen, • können Probleme und Schwachstellen epidemiologischer Beweisführungen benennen und kritisch reflektieren, • können nach Absolvieren des Blockpraktikums die methodischen Vorgehensweisen selbständig in einer statistischen Programmierumgebung praktisch auf die Analyse von Datensätzen anwenden.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!

9	**Verwendbarkeit des Moduls**	Nebenfach Medizinische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Nebenfach Philosophie

1	**Modulbezeichnung** 75330	**Basismodul Philosophie** Basic module: Philosophy	**10 ECTS**
2	Lehrveranstaltungen	Kombiseminar: Kant: Grundlegung zur Metaphysik der Sitten (PS/MS) (SWS) Kombiseminar: Intellektuelle Tugenden (PS/MS) (SWS) Kombiseminar: Gedächtnis, Emotion, Identität - Eine Einführung in die Philosophie der Psychologie (PS/MS) (SWS) Kombiseminar: Albert Camus: Philosophische und literarische Texte (PS/MS) (SWS) Kombiseminar: Descartes, Meditationen über die erste Philosophie PS/MS (2 SWS) Kombiseminar: Platon, Gorgias PS/MS (2 SWS) Kombiseminar: Philosophy of Artificial Intelligence. An Introduction (PS/MS) (SWS) Kombiseminar: Aktuelle Fragen der Angewandten Ethik (PS/MS) (SWS) Kombiseminar: Moralische Rechte (PS/MS) (SWS)	- - - - - - - - -
3	Lehrende	PD Dr. Dagmar Kiesel Prof. Dr. Rosario La Sala Dr. Roberto Redaelli Ufuk Özbe	

4	**Modulverantwortliche/r**	Prof. Dr. Gerhard Ernst
5	**Inhalt**	• Erwerb fundierten Grundlagenwissens eines Teilbereiches der theoretischen oder praktischen Philosophie • Weiterführende systematische Auseinandersetzung mit den für das Teilgebiet zentralen Begriffen • Genaue Diskussion verschiedener in der G
6	**Lernziele und Kompetenzen**	Die Studierenden • machen sich durch die Diskussion der Grundlagen und Grundfragen des jeweiligen Teilbereichs eingehend mit diesem vertraut • erwerben grundlegende Kenntnisse über die Entwicklung des betreffenden Teilgebiets in der Geschichte der Philosophie • erwerben die Fähigkeit komplexe philosophische Inhalte im Seminar zu präsentieren • lernen Hausarbeiten zu schreiben • erwerben im Rahmen des Textseminars die Fähigkeit zentrale Werke der Philosophiegeschichte systematisch zu interpretieren.
7	**Voraussetzungen für die Teilnahme**	Vorherige Absolvierung der Propädeutiken und eines der Grundkurse wird empfohlen

8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Philosophie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Präsentation/Hausarbeit
11	**Berechnung der Modulnote**	Präsentation/Hausarbeit (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Siehe UnivIS

1	**Modulbezeichnung** 75290	**Einführung in die Philosophie** Introduction to philosophy	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Einführung in die Philosophie (Propädeutik I), Gruppe 2 (SWS)	-
3	Lehrende		

4	**Modulverantwortliche/r**	Prof. Dr. Gerhard Ernst
5	**Inhalt**	• Vermittlung von Arbeitstechniken wie Bibliographieren, Exzerpieren, Texte verfassen • Einweisung in die Benutzung der örtlichen Bibliotheken • Vermittlung eines ersten Überblicks in die verschiedenen Teilbereiche der Philosophie • Einführ
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben die Fähigkeit, die Arbeitsmittel und -techniken selbständig zu gebrauchen, die für ihr Philosophiestudium unerlässlich sind • erwerben grundlegende Kenntnisse der philosophischen Begrifflichkeit • lernen Texte auf ihre argumentative Struktur hin zu durchschauen und zu analysieren • gewinnen einen ersten Überblick über die verschiedenen Teilbereich der Philosophie
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Philosophie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (0%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Siehe UnivIS

1	**Modulbezeichnung** 75310	**Grundkurs Praktische Philosophie** Basic course: Practical philosophy	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Grundkurs Praktische Philosophie, Gruppe 2 (SWS) Tutorium: Tutorium Grundkurs Praktische Philosophie (2 SWS) Tutorium: Tutorium Grundkurs Praktische Philosophie (2 SWS) Übung: Übung zum GK Prakt. Philosophie (SWS)	- - - -
3	Lehrende	Ufuk Özbe	

4	**Modulverantwortliche/r**	Prof. Dr. Erasmus Mayr
5	**Inhalt**	• Vermittlung fundierter Grundlagen der Ethik • Systematische Diskussion von Termini wie Moral und Ethik, Autonomie, Glück, freier Wille, Gerechtigkeit • Vermittlung der Kenntnis verschiedener in der Geschichte der Philosophie vertretener Ansä
6	**Lernziele und Kompetenzen**	Die Studierenden • erlangen fundierte Kenntnisse über die Grundlagen und Grundprobleme der Ethik • erwerben Grundkenntnisse über die philosophiegeschichtliche Entwicklung der Ethik • werden in den systematischen Umgang und die Analyse mit zentralen historischen und zeitgenössischen Texten der Ethik eingeführt
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 1
9	**Verwendbarkeit des Moduls**	Nebenfach Philosophie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	schriftlich/mündlich
11	**Berechnung der Modulnote**	schriftlich/mündlich (0%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Siehe UnivIS

Nebenfach Sinologie

1	**Modulbezeichnung** 76003	**Modernes Chinesisch 1** Modern Chinese 1	**10 ECTS**
2	Lehrveranstaltungen	Übung: BA-1 Modernes Chinesisch I (Gruppe 1) (SWS)	-
3	Lehrende	Jingjing Zhang Mengyuan Tian	

4	**Modulverantwortliche/r**	Prof. Dr. Andrea Bréard
5	**Inhalt**	Einführung in die chinesische Sprache und Schrift, wobei ein besonderer Akzent auf den aktiven Schriftzeichenerwerb gelegt wird, ohne die mündliche Kommunikation und die grundlegende Grammatik zu vernachlässigen.
6	**Lernziele und Kompetenzen**	Die Studierenden... ...beherrschen die Pinyin-Umschrift, ...sind zur einfachen Kommunikation befähigt, ...erwerben 300 Schriftzeichen und die in diesen 300Schriftzeichen enthaltenen Radikale, ...haben Sicherheit in den grundlegenden grammatischen Kenntnissen wie Ja-Nein-Fragen, W-Fragen, Possessivpronomen, Satzstellung, Attributpartikel de", Zahlen und Uhrzeiten, Verneinung, Modalverben etc.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Sinologie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten) mündlich (10 Minuten)
11	**Berechnung der Modulnote**	Klausur (70%) mündlich (30%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 120 h Eigenstudium: 180 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 76012	**Modernes Chinesisch 2** Module 2: Modern Chinese 2	**10 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Andrea Bréard
5	**Inhalt**	Der aktive Schriftzeichenerwerb steht weiterhin im Zentrum, wobei mit fortschreitenden Lektionen mehr Themen und damit eine erweiterte Möglichkeit zur mündlichen Kommunikation angeboten werden sowie grammatische Kenntnisse mit zunehmendem Schwierigkeitsgrad zu bewältigen sind.
6	**Lernziele und Kompetenzen**	Die Studierenden... ...erwerben aktiv weitere 300 Schriftzeichen (1. und 2.Semester bis zu 600 Schriftzeichen) und damit Vertrautheit mit den Strukturen der Schriftzeichen, wodurch ein schnelles Erkennen bei der Textlektüre gefördert wird, ...entwickeln eine grundlegende Lesefähigkeit, ...sind sicher im Umgang mit aufbauenden grammatischen Kenntnissen wie Positionswörtern, Verlauf einer Handlung, Vollendung eines Geschehens, präpositionale Konstruktion etc., ...besitzen die Kommunikationsfähigkeit zu erweiterten Themen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Sinologie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich (10 Minuten) Klausur (120 Minuten)
11	**Berechnung der Modulnote**	mündlich (30%) Klausur (70%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 112 h Eigenstudium: 188 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	

Nebenfach Ökonomie

1	**Modulbezeichnung** 74860	**Betriebliches Rechnungswesen I** Cost accounting I	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Betriebliches Rechnungswesen I (2 SWS, WiSe 2023) Übung: Übung 3 BRW I (1 SWS)	5 ECTS -
3	Lehrende	Dr. Ralf Pohl Tobias Mücke Jessica Doms	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Ökonomie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 74810	**Betriebswirtschaftslehre I** Business administration I	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übung 3 BWL I (1 SWS) Vorlesung: Betriebswirtschaftslehre I (4 SWS)	- 5 ECTS
3	Lehrende	Lana Rauf Sebastian Klare Prof. Dr. Matthias Fifka Milena Störmer	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Ökonomie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 74830	**Betriebswirtschaftslehre II** Business administration II	**5 ECTS**
2	Lehrveranstaltungen	Seminar: Abschlussarbeiten-Seminar Betriebswirtschaftslehre (1 SWS) Übung: Übungen zur Vorlesung Betriebswirtschaftslehre II (1 SWS) Vorlesung: Betriebswirtschaftslehre II (4 SWS)	- - 5 ECTS
3	Lehrende	Iman Hamadi Sebastian Klare Prof. Dr. Matthias Fifka	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Ökonomie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 74820	**Einführung in die Volkswirtschaftslehre** Introduction to economics	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Einführung in die Volkswirtschaftslehre (2 SWS, WiSe 2023) Übung: Übung 3 EVWL (1 SWS)	5 ECTS -
3	Lehrende	Prof. Dr. Johannes Paha Andreas Link Michelle Zerrahn Antonia Henneberger	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Ökonomie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 38051	**International Business Ethics I** International business ethics I	**5 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Prof. Dr. Matthias Fifka Milena Störmer
5	**Inhalt**	After the financial crisis and recent corporate scandals at Wirecard, Volkswagen, Sanlu, Deutsche Bank, and many others, the call for ethical management of businesses, especially multinational corporations, has increased dramatically. In this course, we will look at the broad area of business ethics. As a theoretical foundation, ethical theories will be considered and concepts such as sustainability, corporate social responsibility (CSR), and corporate citizenship (CC) will be analyzed. Moreover, in the age of globalization, climate change, and grand challenges such as the coronavirus SARS-CoV-2 pandemic, business increasingly takes place in an environment where political and cultural boundaries that challenge corporate values are constantly crossed. Businesses are constantly confronted with issues such as environmental depletion, poverty, child labor, bribery and corruption, ethnic and religious discrimination, censorship and privacy issues, fair trade, and consumer boycotts, to mention a few. Thus, we will address, e.g., the challenges that organizations are confronted with when trying to do business ethically, what it takes to be a good corporate citizen, and how companies can deal with stakeholder demands.
6	**Lernziele und Kompetenzen**	Students acquire basic knowledge of ethics and business ethics. They are made familiar and learn how to address the challenges that organizations face when trying to do business ethically, be a good corporate citizen, and deal with stakeholder demands. In this lecture, students will: • acquire basic knowledge in ethics and business ethics; • understand important ethical theories and apply them to practical examples; • understand important concepts such as corporate social responsibility and sustainability;

		• learn how to address the challenges that organizations are confronted with when trying to do business ethically; • critically reflect what it takes to be a good corporate citizen; • develop strategies to deal with stakeholder demands; • learn how to implement corporate governance in an organization.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Ökonomie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 45 h Eigenstudium: 105 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Austin, J.; Stevenson, H. & Wei- Skillern, J. (2006) “Social and commercial entrepreneurship: same, different, or both?” Entrepreneurship, Theory and Practice, 30 (1): 1-22. • Crane A. & Matten, D. (2019) “Business ethics: Managing Corporate Citizenship and Sustainability in the Age of Globalization”, 5th Edition, Oxford University Press: Oxford. • Enderle, G. (2000) “Business Ethics in the Intercultural and Global Context: A conceptual framework”, Zeitschrift für Wirtschafts- und Unternehmensethik / Journal for Business, Economics & Ethics, 1(3): 263-285. • Fifka, M. & Loza Adaui, C. (2015) “Managing stakeholders for the sake of business and society”, in O’Riordan, L.; Zmuda, P & Heinemann, S. (Eds.) New Perspectives on Corporate Social Responsibility. Locating the Missing Link, Springer Gabler: Wiesbaden, pp. 71-88. • Friedman, M. (1970) “The Social Responsibility of Business Is to Increase Its Profits.” The New York Times Magazine. • Garriga, E. and Melé, D. (2004) “Corporate Social Responsibility Theories: Mapping the Territory”, Journal of Business Ethics, 53: 51-71 • Grant, C. (1991) “Friedman fallacies”, Journal of Business Ethics, 10: 907-914.

		• Melé, D. (2020) “Business ethics in action: Managing human excellence in organizations”, 2nd Edition, Red Globe Press: London. • Trevino, L.K. and Brown, M.E. (2004) “Managing to be Ethical: Debunking Five Business Ethics Myths”, Academy of Management Executive, 18 (2): 69-81.

1	**Modulbezeichnung** 82070	**Makroökonomie** Macroeconomics	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Christian Merkl Prof. Dr. Claus Schnabel
5	**Inhalt**	• Fragestellungen der Makroökonomie • Grundzüge der Volkswirtschaftlichen Gesamtrechnungen • Entstehung, Verwendung und Verteilung des BIP • Geld und Inflation • Die offene Volkswirtschaft • Wirtschaftswachstum • Langfristiges Gleichgewicht vs. kurzfristige Schwankungen • Gesamtwirtschaftliche Nachfrage • Zusammenwirken von Gesamtangebot und -nachfrage • Von der makroökonomischen Theorie zur Politik
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben fundierte Kenntnisse über Grundfragen, Begrifflichkeit und wirtschaftspolitische Relevanz der Makroökonomie. • verstehen und erklären gesamtwirtschaftliche Prozesse und Phänomene anhand der Arbeitsmaterialien. • können ein einfaches Modell des langfristigen makroökonomischen Gleichgewichts handhaben und darin die Ursachen von Konjunkturschwankungen und die Wirkungsweise von Geld- und Fiskalpolitik abbilden. • beherrschen ein Modell des langfristigen gleichgewichtigen Wirtschaftswachstums und können die wesentlichen Einflussfaktoren des Wachstums identifizieren. • können die vorgestellten Theorien kritisch reflektieren. • sind in der Lage, gesamtwirtschaftliche Entwicklungen einzu-schätzen, wirtschaftspolitische Maßnahmen kritisch zu hinterfragen und Handlungsempfehlungen abzugeben.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 2
9	**Verwendbarkeit des Moduls**	Nebenfach Ökonomie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Lehrbuch: Makroökonomie, N. Gregory Mankiw, 7. Aufl. 2017

1	**Modulbezeichnung** 74840	**Mikroökonomie** Microeconomics	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übung 5 Mikro (1 SWS) Keine Anwesenheitspflicht. Arbeitsaufwand • Vorlesung: 30 Stunden • Übung: 15 Stunden • Literaturstudium: 65 Stunden • Übungsaufgaben: 30 Stunden • Vorbereitung Klausur: 10 Stunden	-
3	Lehrende	Luiza Jarosa Alina Imping Antonia Henneberger	

4	**Modulverantwortliche/r**	Alina Imping
5	**Inhalt**	Inhalt und Qualifikationsziel Die Vorlesung vertieft den mikroökonomischen Stoff der Vorlesung Einführung in die Volkswirtschaftslehre und ergänzt ihn durch zusätzliche Aspekte wie spieltheoretische Erklärungsansätze von Kooperation und Konflikt, Entscheidung bei Unsicherheit, allgemeines Gleichgewicht und asymmetrische Informationen. Ziel der Vorlesung ist vor allem, das methodische Rüstzeug zu vermitteln, das zur Analyse mikroökonomischer Phänomene benötigt wird. Themen: 1) Individuelle Nachfrage und Marktnachfrage 2) Entscheidungen bei Unsicherheit 3) Produktion 4) Kosten der Produktion 5) Gewinnmaximierung und Wettbewerbsangebot 6) Monopol 7) Oligopol 8) Spieltheorie und Wettbewerbsstrategie 9) Märkte für Produktionsfaktoren 10) Märkte mit asymmetrischer Information 11) Allgemeines Gleichgewicht und ökonomische Effizienz
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Voraussetzungen für die Teilnahme Die Kenntnisse des Moduls „Einführung in die Volkswirtschaftslehre“ werden vorausgesetzt.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Ökonomie Bachelor of Science Informatik 20222 Verwendbarkeit des Moduls

		B.A. Ökonomie, Lehramt Gymnasium, B.Sc. Wirtschaftsmathematik, B.Sc. Mathematik (Nebenfach Volkswirtschaftslehre), B.Sc. Physische Geographie (Nebenfach Ökonomie), B.A. Kulturgeographie (1-Fach) (Nebenfach Ökonomie), B.Sc. Psychologie (Nebenfach Ökonomie/ Wirtschaftswissenschaften), M.Sc. Psychologie (Nebenfach Ökonomie/ Wirtschaftswissenschaften)
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) Voraussetzungen für die Vergabe von Leistungspunkten • 4-5 Computerbasierte Tests; das Bestehen der computerbasierten Tests (mindestens 50 % der insgesamt möglichen Punktzahl) ist Voraussetzung für die Teilnahme an der Klausur. (nur für Studierende mit Studienbeginn bis Wintersemester 2018/19) • Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%) Leistungspunkte und Noten • 5 ECTS Punkte • Bewertet: ja (Drittelnoten) • Gewichtung der Prüfungsleistungen in der Benotung: Klausur 100 Prozent, Computerbasierte Tests 0 Prozent
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	Ein Semester Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 75060	**Monetäre Ökonomik** Monetary economics	**5 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Ökonomie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 74870	**Sozialpolitik** Social policy	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Sozialpolitik (2 SWS) Übung: Übungen zur Vorlesung "Sozialpolitik" (1 SWS)	5 ECTS 0 ECTS
3	Lehrende	Prof. Dr. Christine Binzel Andreas Link	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Ökonomie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 74790	**Strategisches Management** Strategic management	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Strategisches Management (3 SWS)	5 ECTS
3	Lehrende	Dr. Cristian Rolando Loza Adaui	

4	**Modulverantwortliche/r**	Dr. Cristian Rolando Loza Adaui
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Ökonomie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Nebenfach EEI: Allgemeine Elektrotechnik

1	**Modulbezeichnung** 95610	**Werkstoffkunde für EEI** Materials science for EECE	**2,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Werkstoffkunde für Studierende der Elektrotechnik (EEI) (2 SWS)	2,5 ECTS
3	Lehrende	Prof. Dr.-Ing. Peter Wellmann	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Peter Wellmann
5	**Inhalt**	Vorlesung Werkstoffkunde (EEI, 1. Semester) 1.Streifzug durch die Werkstoffkunde 2.Kristalle Aufbau der Materie, kristalline Ordnung, dichteste Kugelpackung, Kristallbaufehler 3.Kristallbindung & Phasendiagramme Typen von Atombindungen im Kristallgitterverband, Schmelz- und Zersetzungstemperaturen von reinen Stoffen, von binären und von ternären Systemen, Phasendiagramme von Legierungen 4.Mechanische Eigenschaften von Festkörpern Material-Kenngrößen, klassischer" Zugversuch und Spannungs-Dehnungs-Diagramm, Plastische Verformung, Kriechverhalten, Rissbildung, Begriff der Härte. Ergänzung (kein Prüfungsstoff) Formgedächtnis-Legierung 5.Elektrische und Thermische Eigenschaften der Materie Grundlagen des elektrischen Ladungstransportes in Festkörpern, Mikroskopisches Bild des Ohmschen Gesetzes, elektrischer Transport in Metallen, Grundlagen des Wärmetransportes, Kühlung elektronischer Baugruppen 6.Metalle Elektrische Kabel und Leitungen, Lote, Kontakt- und Thermospannung, Peltiereffekt 7.Halbleiter Halbleiter-Grundlagen, Halbleitermaterial Silizium, Kristallzüchtung und Wafer-Herstellung, pn-Diode, µ-Elektronik, das Si-SiO2-Interface, Optoelektronik, Verbindungshalbleiter, Lichterzeugung, Lichtabsorption, Leucht- und Laserdiode, Glasfasern, Ergänzung (kein Prüfungsstoff) Farbdisplays, Amorphes und polykristallines Silizium für Solarzellen und TFTs, Photodiode und Solarzelle 8.Dielektrika Grundlagen, Materialkenngrößen, Materialklassen, Herstellungsverfahren, Isolatoren, Piezo- und Ferroelektrizität 9.Magnetismus Physikalische und materialwissenschaftliche Grundlagen, Materialien, Spulen und Transformatoren, Datenspeicherung 10.Supraleiter Physikalische Grundlagen, Tief- und Hochtemperatur-Supraleiter, Typische Materialien, Anwendungen
6	**Lernziele und Kompetenzen**	Die Studierenden erwerben fundierte Kenntnisse über Materialeigenschaften.

7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Allgemeine Elektrotechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten) Die Vorlesung des Lehrstuhls WW6 wird im Format "**Flipped Classroom**" durchgeführt (synchrone Lerneinheiten im Hörsaal & asynchrone Lerneinheiten über Studon: https://www.studon.fau.de/studon/goto.php?target=crs_359194) Die (Teil-)Prüfung WW6 findet als **elektronische Klausur** (maximale Punktzahl = 300) statt. Die elektronische Klausur enthält teilweise Multiple Choice Fragen. Es gilt: Jede Antwortmöglichkeit wird bei richtiger Beantwortung mit der zugewiesenen Punktzahl bewertet; falsche Beantwortung geht innerhalb der Frage mit negativen Punkten ein. Es werden alle Punkte der Antwortmöglichkeiten addiert. Es gibt keine Negativpunkte für falsch markierte Aufgaben.
11	**Berechnung der Modulnote**	Klausur (100%) Für die Klausur können im Rahmen des Flipped-Classroom Konzeptes bis zu 30 **Bonuspunkte** für die Abschlussklausur erworben werden indem an 5 von 7 Präsenzterminen die Widerholungsfragen (= Kickoff-Polls) zum Beginn der Veranstaltung erfolgreich beantwortet werden (50-75% richtige Antworten: Bonus = 15 Punkte, >75% richtige Antworten: Bonus = 30 Punkte). Hinweis: Als Vorbereitung für die Kickoff-Polls in den Präsenzphasen wir die die Teilnahme am **eTutorium** (Kurs der Virtuellen Hochschule Bayern) empfohlen: https://kurse.vhb.org/VHBPORTAL/kursprogramm/kursprogramm.jsp --> WS xx/xx --> Ingenieurwissenschaften --> Elektrotechnik/Elektronik und Informationstechnik --> Werkstoffkunde für die Elektrotechnik (bitte die Studiengang-Auswahl beachten !)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 45 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	P. Wellmann - Materialien der Elektronik und Energietechnik - Halbleiter, Graphen, Funktionale Materialien, Springer-Verlag

Nebenfach EEI: Automatisierungstechnik

1	**Modulbezeichnung** 92580	**Grundlagen der Elektrotechnik III** Foundations of electrical engineering III	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übungen zu Grundlagen der Elektrotechnik III (2 SWS) Vorlesung: Grundlagen der Elektrotechnik III (2 SWS) Tutorium: Tutorium zu Grundlagen der Elektrotechnik III (0 SWS)	- 5 ECTS -
3	Lehrende	Prof. Dr.-Ing. Philipp Beckerle Daniel Andreas	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Philipp Beckerle
5	**Inhalt**	• Umfang und Bedeutung der elektrischen Messtechnik • Die Grundlagen des Messens • Fourier-Transformation • Laplace-Transformation • Netzwerkanalyse im Zeit- und Laplace-Bereich • Übertragungsfunktion und Bode-Diagramm • Nichtlineare Bauelemente, Schaltungen und Systeme • Operationsverstärker • Messverstärker • Messfehler • Messung von Gleichstrom und Gleichspannung • Ausschlagbrücken • Abgleichbrücken, Messung von elektrischen Impedanzen
6	**Lernziele und Kompetenzen**	Die Studierenden • ordnen die behandelten Verfahren gemäß ihrer Eignung für spezifische Probleme (Zeit-/Frequenzbereich, Linear/ Nichtlinear) ein. • wählen geeignete Verfahren zur Analyse elektrischer Netzwerke aus und wenden diese an. • interpretieren die Ergebnisse und zeigen Zusammenhänge zwischen den Lösungsverfahren auf. • kennen einfache Grundschaltungen mit Operationsverstärkern und sind in der Lage, diese zu analysieren. • kennen die behandelten Messschaltungen und ihre Einsatzmöglichkeiten. • analysieren Brückenschaltungen. • wenden grundlegende Konzepte der Messfehlerrechnung auf Messschaltungen an. • reflektieren selbstständig den eigenen Lernprozess und nutzen die Präsenzzeit zur Klärung der erkannten Defizite.
7	**Voraussetzungen für die Teilnahme**	Grundlagen der Elektrotechnik I und II
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!

9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Automatisierungstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Lehrbuch: Elektrische Messtechnik", R. Lerch, 7. Aufl. 2016, Springer-Verlag Übungsbuch: Elektrische Messtechnik Übungen", R. Lerch, M. Kaltenbacher, F. Lindinger, A. Sutor, 2. Aufl. 2005, Springer-Verlag

1	**Modulbezeichnung** 97060	**Regelungstechnik B (Zustandsraummethoden)** Control engineering B (State-space methods)	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Regelungstechnik B (Zustandsraummethoden) (4 SWS)	5 ECTS
3	Lehrende		

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Knut Graichen
5	**Inhalt**	Die Vorlesung vermittelt die Grundlagen zur Beschreibung und Untersuchung von linearen dynamischen Systemen mit mehreren Ein- und Ausgangsgrößen im Zustandsraum sowie den zustandsraumbasierten Regler- und Beobachterentwurf. Die Inhalte der Vorlesung sind: • Motivation der Zustandsraumbetrachtung dynamischer Systeme in der Regelungstechnik • Zustandsraumdarstellung dynamischer Systeme und deren Vereinfachung durch Linearisierung • Analyse linearer und zeitinvarianter Systeme: Stabilität, Steuerbarkeit, Beobachtbarkeit, Zusammenhang mit Ein-/Ausgangsbetrachtung • Auslegung von linearen Zustandsreglern für lineare Eingrößensysteme • Erweiterte Regelkreisstrukturen, insbesondere Vorsteuerung und Störgrößenkompensation • Entwurf von Zustands- und Störgrößenbeobachtern und Kombination mit Zustandsreglern (Separationsprinzip)
6	**Lernziele und Kompetenzen**	Die Studierenden können • die Vorzüge der Zustandsraumbetrachtung im Vergleich zur Ein-/Ausgangsbetrachtung darlegen. • für dynamische Systeme die Zustandsgleichungen aufstellen und durch Linearisierung vereinfachen. • für LZI-Systeme die Zustandsgleichungen in Normalformen transformieren. • Stabilität, Steuer- und Beobachtbarkeit von Zustandssystemen definieren und LZI-Systeme daraufhin untersuchen. • ausführen, wie diese Eigenschaften mit den Eigenwerten und Nullstellen von LZI-Zustandssystemen zusammenhängen. • den Aufbau einer Zwei-Freiheitsgrade-Zustandsregelung angeben und die Zweckbestimmung ihrer einzelnen Komponenten erläutern. • realisierbare Vorsteuerungen zur Einstellung des Sollverhaltens entwerfen. • Zielstellung und Aufbau eines Zustandsbeobachters erläutern. • diesen zu einem Störbeobachter erweitern und Störaufschaltungen zur Kompensation von Dauerstörungen konzipieren. • beobachterbasierte Zustandsregelungen durch Eigenwertvorgabe entwerfen.

		• die Vorlesungsinhalte auf verwandte Problemstellungen übertragen und sich die Zustandsraummethoden der Regelungstechnik selbständig weiter erschließen.
7	**Voraussetzungen für die Teilnahme**	Vektor- und Matrizenrechnung sowie Grundlagen der Regelungstechnik (klassische Frequenzbereichsmethoden; kann auch parallel gehört werden, siehe Regelungstechnik A)
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Automatisierungstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) Die Summe der in den Online-Tests erzielten Punktzahl wird zu max. 10% auf die Klausurpunktzahl angerechnet. Hiermit ist eine Verbesserung der Klausurbewertung um bis zu 0,7 Notenpunkte möglich. Die Anrechnung erfolgt nur, wenn Sie die Prüfung an sich mit der Mindestnote 4,0 bestanden haben. Der Bonus kann nur einmal im Prüfungszeitraum der Vorlesung angerechnet werden, entweder zum Haupttermin nach Vorlesungsende oder zum Nachholtermin im Folgesemester, wenn der Haupttermin nicht wahrgenommen wurde.
11	**Berechnung der Modulnote**	Klausur (100%) Die Summe der in den Online-Tests erzielten Punktzahl wird zu max. 10% auf die Klausurpunktzahl angerechnet. Hiermit ist eine Verbesserung der Klausurbewertung um bis zu 0,7 Notenpunkte möglich. Die Anrechnung erfolgt nur, wenn Sie die Prüfung an sich mit der Mindestnote 4,0 bestanden haben. Der Bonus kann nur einmal im Prüfungszeitraum der Vorlesung angerechnet werden, entweder zum Haupttermin nach Vorlesungsende oder zum Nachholtermin im Folgesemester, wenn der Haupttermin nicht wahrgenommen wurde.
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• C.T. Chen. Control System Design, Pond Woods Press, 1987 • O. Föllinger. Regelungstechnik: Einführung in die Methoden und ihre Anwendung. 8. Auflage, Hüthig, 1994 • H. Geering. Regelungstechnik, 6. Auflage, Springer, 2004 • T. Kailath. Linear Systems, Prentice Hall, 1980 • G. Ludyk. Theoretische Regelungstechnik 1, Springer, 1995 • D.G. Luenberger. Introduction to Dynamic Systems, John Wiley & Sons, 1979 • J. Lunze. Regelungstechnik 1, 12. Auflage, Springer, 2020 • J. Lunze. Regelungstechnik 2, 10. Auflage, Springer, 2020 • L. Padulo, M.A. Arbib. System Theory, W.B. Saunders Company, 1974

		• W.J. Rugh. Linear System Theory 2, Prentice Hall, 1996

Nebenfach EEI: Elektrische Energie- und Antriebstechnik

1	**Modulbezeichnung** 92542	**Grundlagen der elektrischen Energieversorgung** Fundamentals of electrical energy supply	**4 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Matthias Luther
5	**Inhalt**	Ausgehend von der Bedeutung und den Eigenschaften der elektrischen Energie wird der Aufbau von Energieversorgungssystemen beschrieben und die wichtigsten Systemelemente im Überblick vorgestellt. Die Grundlagen der Wechselstromtechnik werden erarbeitet und die gebräuchlichen Koordinatentransformationen für Dreiphasensysteme und ihre wechselseitigen Zusammenhänge behandelt. Mit ihrer Hilfe werden die Hauptelemente symmetrischer Drehstromnetzwerke transformiert und die Modellierung und Berechnung von Drehstromnetzen im symmetrischen und unsymmetrischen Betrieb vorgestellt. Ausführlich folgen die Leistungsverhältnisse in Elektroenergiesystemen als Grundlage für deren Auslegung und Betrieb, einschließlich nichtkosinusförmiger periodischer Dreiphasensysteme. Den Abschluß bilden Fragen der wirtschaftlichen elektrischen Energieversorgung.
6	**Lernziele und Kompetenzen**	Die Studierenden können grundlegende und praxisnahe Herausforderungen der elektrischen Energieversorgung einschätzen und die Zusammenhänge über die technischen Grundlagen und Begrifflichkeiten erläutern. Sie bauen auf ihren Vorkenntnissen (aus Grundlagen der Elektrotechnik) auf und können die Berechnungsgrundlagen für die elektrische Energieversorgung anwenden. Sie sind in der Lage, die notwendigen physikalischen Grundlagen, mathematischen Verfahren und Werkzeuge anhand von praxisnahen Beispielen darzustellen und anzuwenden: • Übergang vom Wechselstrom- zum Drehstromsystem mit Berechnung der verschiedenen Leistungsarten in ein- und dreiphasigen Systemen • Betriebsmöglichkeiten hybrider Systeme • Vier- und Achtpoltheorie sowie unterschiedlicher Modaltransformationen, um symmetrische und unsymmetrische Betriebszustände in Drehstromsystemen zu analysieren • Berechnungsverfahren zur Kenngrößenbestimmung von Leitungen Weiterhin können die Studierenden Grundkenntnisse und Berechnungsverfahren für alle relevanten Betriebsmittel und Komponenten der elektrischen Energieversorgung anwenden. Sie können die Grundzüge der elektrischen Energiewirtschaft und Netzbetriebsführung erklären und die wirtschaftlichen und operativen

		Prozesse in der Elektrischen Energieversorgung einordnen. Damit sind sie in der Lage, ihr Wissen in weiterführendenden energietechnischen Vorlesungen auszubauen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Elektrische Energie- und Antriebstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Skript zur Vorlesung • Herold, Gerhard: Elektrische Energieversorgung I. Drehstromsysteme - Leistungen - Wirtschaftlichkeit. 3. Aufl. Wilburgstetten: J. Schlembach Fachverlag, 2011 - 400 Seiten. ISBN 3-935340-69-9

1	**Modulbezeichnung** 92560	**Grundlagen der Elektrotechnik I** Foundations of electrical engineering I	**7,5 ECTS**
2	Lehrveranstaltungen	Übung: Übung GET I (2 SWS) Vorlesung: Grundlagen der Elektrotechnik I (4 SWS)	- 7,5 ECTS
3	Lehrende	Gregor Hofmann Prof. Dr. Bernd Witzigmann	

4	**Modulverantwortliche/r**	Prof. Dr. Bernd Witzigmann
5	**Inhalt**	Diese Vorlesung bietet einen Einstieg in die physikalischen Grundlagen der Elektrotechnik und Elektronik. Ausgehend von beobachtbaren Kraftwirkungen zwischen Ladungen und zwischen Strömen wird der Begriff des elektrischen und magnetischen Feldes eingeführt. Mit den daraus abgeleiteten integralen Größen Spannung, Strom, Widerstand, Kapazität und Induktivität wird das Verhalten der passiven Bauelemente diskutiert. Am Beispiel der Gleichstromschaltungen werden die Methoden der Netzwerkanalyse eingeführt und Fragen nach Wirkungsgrad und Zusammenschaltung von Quellen untersucht. Einen Schwerpunkt bildet das Faradaysche Induktionsgesetz und seine Anwendungen. Die Bewegungsinduktion wird im Zusammenhang mit den Drehstromgeneratoren betrachtet, die Ruheinduktion wird sehr ausführlich am Beispiel der Übertrager und Transformatoren diskutiert. Einen weiteren Schwerpunkt bildet die Behandlung zeitlich periodischer Vorgänge. Die komplexe Wechselstromrechnung bei sinusförmigen Strom- und Spannungsformen wird ausführlich behandelt. 1. Physikalische Grundbegriffe 2. Das elektrostatische Feld 3. Das stationäre elektrische Strömungsfeld 4. Einfache elektrische Netzwerke 5. Das stationäre Magnetfeld 6. Das zeitlich veränderliche elektromagnetische Feld 7. Wechselspannung und Wechselstrom
6	**Lernziele und Kompetenzen**	Nach der Teilnahme an den Modulveranstaltungen sind die Studierenden in der Lage: • den Begriff des Feldes zu verstehen, • Gleich- und Wechselstromschaltungen mit Widerständen, Kapazitäten, Induktivitäten und Transformatoren zu entwickeln, • Schwingkreise und Resonanzerscheinungen zu analysieren, • Energie- und Leistungsberechnungen durchzuführen, • Schaltungen zur Leistungsanpassung und zur Blindstromkompensation zu bewerten, • das Drehstromsystem zu verstehen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!

9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Allgemeine Elektrotechnik Bachelor of Science Informatik 20222 Nebenfach EEI: Elektrische Energie- und Antriebstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• M. Albach, Elektrotechnik, Pearson Verlag • Manfred Albach: Grundlagen der Elektrotechnik I und II, Pearson-Verlag • Übungsaufgaben mit Lösungen auf der Homepage • Optional: Übungsbuch, Pearson-Verlag

1	**Modulbezeichnung** 92570	**Grundlagen der Elektrotechnik II** Foundations of electrical engineering II	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Klaus Helmreich
5	**Inhalt**	Diese Veranstaltung stellt den zweiten Teil einer 3-semestrigen Lehrveranstaltung über Grundlagen der Elektrotechnik für Studierende der Elektrotechnik, Elektronik und Informationstechnik im Grundstudium dar. Inhalt ist die Analyse elektrischer Grundschaltungen und Netzwerke aus konzentrierten Bauelementen bei sinus- und nichtsinusförmiger harmonischer Erregung. Nach kurzer Einführung in die komplexe Wechselstromrechnung und den Umgang mit elementaren elektrischen Bauelementen werden zunächst Spannungs- und Stromquellen und ihre Zusammenschaltung mit einer Last sowie die Leistungsübertragung von der Quelle zur Last betrachtet. Nach Herleitung und beispielhafter Anwendung von Methoden und Sätzen zur Berechnung und Vereinfachung elektrischer Schaltungen (Überlagerungssatz, Reziprozitätstheorem, äquivalente Schaltungen, Miller-Theorem etc.) werden zunächst 2-polige Netzwerke analysiert und in einem weiteren Kapitel dann allgemeine Verfahren zur Netzwerkanalyse wie das Maschenstromverfahren und das Knotenpotenzialverfahren behandelt. Die Berechnung der verallgemeinerten Eigenschaften von Zweipolfunktionen bei komplexen Frequenzen führt im verlustlosen Fall zur schnellen Vorhersagbarkeit des Frequenzverhaltens und zu elementaren Verfahren der Schaltungssynthese. Der nachfolgende Teil über mehrpolige Netzwerke konzentriert sich nach der Behandlung von allgemeinen Mehrtoren auf 2-Tore und ihr Verhalten, ihre verschiedenen Möglichkeiten der Zusammenschaltung und die zweckmäßige Beschreibung in verschiedenen Matrixdarstellungen (Impedanz-, Admittanz-, Ketten-, Hybridmatrix). Das Übertragungsverhalten von einfachen und verketteten Zweitoren wird am Beispiel gängiger Filterarten durchgesprochen und das Bode-Diagramm zur schnellen Übersichtsdarstellung eingeführt. Nach allgemeiner Einführung der Fourierreihenentwicklung periodischer Signale wird die Darstellung von nicht sinusförmigen periodischen Erregungen von Netzwerken mittels reeller und komplexer Fourierreihen und die stationäre Reaktion der Netzwerke auf diese Erregung behandelt. Als mögliche Ursache für nichtsinusförmige Ströme und Spannungen in Netzwerken werden nichtlineare Zweipole mit ihren Kennlinienformen vorgestellt und auf die Berechnung des erzeugten Oberwellenspektrums eingegangen.
6	**Lernziele und Kompetenzen**	Die Studierenden

		• besitzen spezialisiertes und vertieftes Wissen über die Umformung, Analyse und Synthese von einfachen und umfangreicheren Netzwerken bei sinus- und nichtsinusförmiger Erregung in komplexer Darstellung. • können die im Inhalt beschriebenen Verfahren und Methoden der Netzwerkanalyse erklären und auf Schaltungsbeispiele anwenden. • können Verfahren der Netzwerkanalyse hinsichtlich des Rechenaufwandes beurteilen und vergleichen.
7	**Voraussetzungen für die Teilnahme**	• Grundlagen der Elektrotechnik 1 • Mathematik I • Mathematik II (begleitend)
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Allgemeine Elektrotechnik Bachelor of Science Informatik 20222 Nebenfach EEI: Elektrische Energie- und Antriebstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Elektrotechnik, Albach, M., 2011. Grundlagen der Elektrotechnik - Netzwerke, Schmidt, L.-P., Schaller, G., Martius, S., 2013. (bisher: Grundlagen der Elektrotechnik 3, Schmidt, L.-P., Schaller, G., Martius, S., 2006.

Nebenfach EEI: Informationstechnik

1	**Modulbezeichnung** 392436	**Grundlagen der Nachrichtenübertragung** Fundamentals of communications	**5 ECTS**
2	Lehrveranstaltungen	Übung: Ergänzungen und Übungen zu Nachrichtentechnische Systeme - Übertragungstechnik (1 SWS) Vorlesung: Nachrichtentechnische Systeme - Übertragungstechnik (3 SWS)	- -
3	Lehrende	Prof. Dr.-Ing. Robert Schober Andreas Feder Prof. Dr. Jörn Thielecke	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Robert Schober Dr.-Ing. Clemens Stierstorfer
5	**Inhalt**	• Einführung und Grundbegriffe • Quellensignale und deren Modellierung • Übertragungskanäle und deren Modellierung • Analoge Modulationsverfahren • Pulscodemodulation • Grundbegriffe der Informationstheorie • Digitale Übertragung
6	**Lernziele und Kompetenzen**	Die Studierenden beschreiben die Aufgaben nachrichtentechnischer Systeme. Sie beschreiben und modellieren Signale mathematisch mit Zufallsprozessen und können diese in den Frequenzbereich transformieren. Sie rechnen lineare Größen in logarithmische Darstellungen um (und zurück) und verwenden die Pegelgrößen sicher. Die Studierenden analysieren analoge Quellensignale, kennen und nutzen dabei die Kenngrößen und Annahmen bzgl. Bandbegrenzung, Spitzenwertbegrenzung usw. Sie unterscheiden analoge und digitale Quellensignale und beschreiben letztere ebenso anhand der üblichen Kenngrößen. Die Studierenden erläutern die Definition des Übertragungskanals sowie mögliche Ursachen für Signalverzerrungen und andere Störeinflüsse. Sie beschreiben den Kanal in äquivalenten komplexen Basisband, insbesondere beschreiben und analysieren sie die Ausbreitung von Signalen bei der Funkübertragung sowie auf Kabeln mit den dort auftretenden Effekten (z.B. Mehrwegeausbreitung, Dämpfung usw.). Sie verwenden additives weißes Rauschen zur Modellierung physikalischer Rauschprozesse in Zeit- und Frequenzbereich. Ebenso verwenden und analysieren die Modelle des AWGN-Kanals und des frequenzselektiven Schwundkanals. Sie bewerten Übertragungsverfahren anhand der Kriterien Leistungseffizienz und Bandbreiteneffizienz. Die Studierenden analysieren und beschreiben mathematisch die gängigen Amplitudenmodulationsverfahren (Ein- und Zweiseitenbandmodulation, Quadraturamplitudenmodulation) in Zeit- und Frequenzbereich. Dies gilt ebenso für die Frequenzmodulation. Sie bewerten diese Modulationsverfahren im Leistungs-Bandbreiten-Diagramm und analysieren den Einfluss von additiven Störern.

		Sie beschreiben die Grundstrukturen der zugehörigen Empfänger, insbesondere des Überlagerungsempfängers. Die Studierenden beschreiben den Übergang von analogen zu digitalen Signalen und analysieren die Effekte von Abtastung und Quantisierung. Sie untersuchen die Auswirkungen von Kompandierung bei der Quantisierung sowie die Anforderungen an die differentielle Pulscodemodulation. Die Studierenden verwenden das Shannon'sche Informationsmaß, Quellencodierungstheorem und die wechselseitige Information zur matehmatischen Beschreibung der Nachrichtenübertragung über gestörte Kanäle. Sie erklären das Kanalcodierungstheorem und analysieren im Detail den AWGN-Kanal und seine Varianten bzgl. informationstheoretische Größen. Die Studierenden erklären die digitale Pulsamplitudenmodulation und analysieren die zugehörigen Sender, die Signale sowie die kohärente Demodulation in Zeit- und Frequenzbereich. Sie ermitteln die Fehlerwahrscheinlichkeit und nutzen dazu das Gaußsches Fehlerintegral und die Error Function. Sie bewerten die digitalen Übertragungsverfahren im Leistungs-Bandbreiten-Diagramm. Die Studierenden verstehen die Motivation für den Einsatz von Kanalcodierung bei digitaler Übertragung.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Informationstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Huber, J.: Skriptum zur Vorlesung Nachrichtenübertragung. 1997. • Kammeyer, K.D.: Nachrichtenübertragung. Teubner, Stuttgart, 2.Aufl., 1996. • Haykin, S.: Communication Systems. John Wiley & Sons, Inc., New York, 1994.

1	**Modulbezeichnung** 92681	**Signale und Systeme I** Signals and systems 1	**5 ECTS**
2	Lehrveranstaltungen	Tutorium: Tutorium zu Signale und Systeme I (TUTSISY I) (1 SWS) Vorlesung: Signale und Systeme I (2 SWS) Übung: Übung zu Signale und Systeme I (2 SWS)	- - -
3	Lehrende	Maximiliane Gruber Prof. Dr.-Ing. Andre Kaup Frank Sippel	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Andre Kaup
5	**Inhalt**	*Kontinuierliche Signale* Elementare Operationen, Delta-Impuls, Energie und Leistung, Skalarprodukt und Orthogonalität, Faltung und Korrelation *Fourier-Transformation* Definition, Symmetrien, inverse Transformation, Sätze und Korrespondenzen *Laplace-Transformation* Definition, Eigenschaften und Sätze, Inverse Transformation, Korrespondenzen *Kontinuierliche LTI-Systeme im Zeitbereich* Impulsantwort, Sprungantwort, Beschreibung durch Differentialgleichungen, Direktformen, Zustandsraumdarstellung, äquivalente Zustandsraumdarstellungen, Transformation auf Diagonalform *Kontinuierliche LTI-Systeme im Frequenzbereich* Eigenfunktionen, Systemfunktion und Übertragungsfunktion, Verkettung von LTI-Systemen, Zustandsraumbeschreibung im Frequenzbereich *Kontinuierliche LTI-Systeme mit Anfangsbedingungen* Lösung mit der Laplace-Transformation, Lösung über die Zustandsraumbeschreibung, Zusammenhang zwischen Anfangswert und Anfangszustand *Kontinuierliche LTI-Systeme mit speziellen Übertragungsfunktionen* Reellwertige Systeme, verzerrungsfreie Systeme, linearphasige Systeme, minimalphasige Systeme und Allpässe, idealer Tiefpass und idealer Bandpass *Kausalität und Hilbert-Transformation* Kausale kontinuierliche LTI-Systeme, Hilbert-Transformation, analytisches Signal *Stabilität und rückgekoppelte Systeme* Übertragungsstabilität, kausale stabile kontinuierliche LTI-Systeme, Stabilitätskriterium von Hurwitz, rückgekoppelte Systeme *Abtastung und periodische Signale* Delta-Impulskamm und seine Fourier-Transformierte, Fourier-Transformierte periodischer Signale, Abtasttheorem, ideale und nichtideale Abtastung und Rekonstruktion, Abtastung im Frequenzbereich

6	**Lernziele und Kompetenzen**	*Die Studierenden* • analysieren kontinuierliche Signale mit Hilfe der Fourier- und Laplace-Transformation • bestimmen die Impulsantwort, Direktformen und Zustandsraumdarstellung für kontinuierliche lineare zeitinvariante Systeme • berechnen System- und Übertragungsfunktionen für kontinuierliche lineare zeitinvariante Systeme • analysieren die Eigenschaften von kontinuierlichen linearen zeitinvarianten Systemen aufgrund der Zeit- und Frequenzbereichsbeschreibung • stufen kontinuierliche lineare zeitinvariante Systeme an-hand ihrer Eigenschaften Verzerrungsfreiheit, Linearphasigkeit und Minimalphasigkeit ein • bewerten Kausalität und Stabilität von kontinuierlichen linearen zeitinvarianten Systemen • beurteilen die Effekte und Grenzen einer Abtastung von kontinuierlichen Signalen
7	**Voraussetzungen für die Teilnahme**	Modul Grundlagen der Elektrotechnik I+II" *oder* Module Einführung in die IuK" sowie Elektronik und Schaltungstechnik"
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Informationstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	B. Girod, R. Rabenstein, A. Stenger, Einführung in die Systemtheorie", Wiesbaden: Teubner-Verlag, 2005

1	**Modulbezeichnung** 92682	**Signale und Systeme II** Signals and systems 2	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Andre Kaup
5	**Inhalt**	*Diskrete Signale* Elementare Operationen und Eigenschaften, spezielle diskrete Signale, Energie und Leistung, Skalarprodukt und Orthogonalität, Faltung und Korrelation *Zeitdiskrete Fourier-Transformation (DTFT)* Definition, Beispiele, Korrespondenzen, inverse zeitdiskrete Fourier-Transformation, Eigenschaften und Sätze *Diskrete Fourier-Transformation (DFT)* Definition, Beispiele, Korrespondenzen, Eigenschaften und Sätze,Faltung mittels der diskreten Fourier-Transformation, Matrixschreibweise, schnelle Fourier-Transformation (FFT) *z-Transformation* Definition, Beispiele, Korrespondenzen, inverse z-Transformation, Eigenschaften und Sätze *Diskrete LTI-Systeme im Zeitbereich* Beschreibung durch Impulsantwort und Faltung, Beschreibung durch Differenzengleichungen, Beschreibung durch Zustandsraumdarstellung *Diskrete LTI-Systeme im Frequenzbereich* Eigenfolgen, Systemfunktion und Übertragungsfunktion, Verkettung von LTI-Systemen, Zustandsraumbeschreibung im Frequenzbereich *Diskrete LTI-Systeme mit speziellen Übertragungsfunktionen* Reellwertige Systeme, verzerrungsfreie Systeme, linearphasige Systeme, minimalphasige Systeme und Allpässe, idealer Tiefpass und ideale Bandpässe, idealer Differenzierer *Kausale diskrete LTI-Systeme und Hilbert-Transformation* Kausale diskrete LTI-Systeme, Hilbert-Transformation für periodische Spektren, analytisches Signal und diskreter Hilbert-Transformator *Stabilität diskreter LTI-Systeme* BIBO-Stabilität, kausale stabile diskrete Systeme, Stabilitätskriterium für Systeme N-ter Ordnung *Beschreibung von Zufallssignalen* Erwartungswerte, stationäre und ergodische Zufallsprozesse, Autokorrelations- und Korrelationsfunktion, Leistungsdichtespektrum, komplexwertige Zufallssignale *Zufallssignale und LTI-Systeme* Verknüpfung von Zufallssignalen, Reaktion von LTI-Systemen auf Zufallssignale, Wienerfilter
6	**Lernziele und Kompetenzen**	Die Studierenden • analysieren diskrete Signale mit Hilfe der zeitdiskreten Fourier-Transformation und berechnen deren diskrete Fourier-Transformation

		• bestimmen die Impulsantwort, Direktformen und Zustandsraumdarstellung für diskrete lineare zeitinvariante Systeme • berechnen System- und Übertragungsfunktionen für diskrete lineare zeitinvariante Systeme • analysieren die Eigenschaften von diskreten linearen zeitinvarianten Systemen aufgrund der Zeit- und Frequenzbereichsbeschreibung • stufen diskrete lineare zeitinvariante Systeme anhand ihrer Eigenschaften Verzerrungsfreiheit, Linearphasigkeit und Minimalphasigkeit ein • bewerten Kausalität und Stabilität von diskreten linearen zeitinvarianten Systemen • bewerten diskrete Zufallssignale durch Berechnung von Erwartungswerten und Korrelationsfunktionen • beurteilen die wesentlichen Effekte einer Filterung von diskreten Zufallssignalen durch diskrete lineare zeitinvariante Systeme
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Informationstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Nebenfach EEI: Mikroelektronik

1	**Modulbezeichnung** 96590	**Entwurf integrierter Schaltungen I** Design of integrated circuits I	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Entwurf Integrierter Schaltungen I (2 SWS) Übung: Übungen zu Entwurf Integrierter Schaltungen I (SWS)	5 ECTS -
3	Lehrende	Prof. Dr.-Ing. Sebastian Sattler Florian Deeg Tobias Rumpel	

4	**Modulverantwortliche/r**	Peter Meisel Prof. Dr.-Ing. Sebastian Sattler
5	**Inhalt**	Es wird in die Grundlagen des integrierten digitalen Schaltungsentwurfes auf Basis von CMOS eingeführt. Ausgehend vom MOS Transistor wird die Complementäre Logik erklärt und auf gängige statische und dynamische Schaltelemente und ihre Erweiterungen auf hochintegrierte Schaltungen bis 0.13µm eingegangen. • Digitaler IC Entwurf für Deep Submicron • MOS Transistor • Herstellung, Layout und Simulation • MOS Inverterschaltung • Statische CMOS Gatter-Schaltungen • Entwurf von Logik mit hoher Schaltrate • Transfer-Gatter und dynamische Logik • Entwurf von Speichern • Zusätzliche Themen des Speicherentwurfs Content It introduces students to the basics of digital integrated circuit design in CMOS. Starting from the MOS transistor, complementary logic is explained. Common static and dynamic switching elements are discussed as well as their extensions to large scale integrated circuits (0.18µm-0.13µm). • Deep Submicron Digital IC Design • MOS Transistor • Fabrication, Layout and Simulation • MOS Inverter Circuits • Static CMOS Gate-Circuits • Design of Logic with High Switching Rate • Transfer-Gates and Dynamic Logic • Design of Memory • Additional Topics of Memory Design
6	**Lernziele und Kompetenzen**	Verstehen • Die Studierenden gewinnen einen Überblick über existierende Integrationstechnologien und Entwurfsmethodiken für Integrierte Schaltungen in 0,18µm und 0,13µm CMOS. Dabei verstehen die Studierenden auch die Zusammenhänge zwischen technischen und wirtschaftlichen Gesichtspunkten der Halbleiterfertigung.

		Evaluieren (Beurteilen) • Die Studierenden analysiert das Verhalten von MOS/CMOS-Transistoren. Daneben können sie verschiedene statische und dynamische digitale Schaltungsstrukturen auf Transistorebene bewerten. Learning objectives and competencies: Understand • gain an overview of existing integration technologies and integrated circuit design techniques in CMOS (0.18µm-0.13µm), understanding technical and economic aspects of semiconductor manufacturing. Evaluate (Assess) • Analyze the behavior of MOS / CMOS transistors and evaluate various static and dynamic digital circuit structures at transistor level.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Mikroelektronik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	Literatur: D. A. Hodges, H. G. Jackson, R. A. Saleh, Analysis and Design of Digital Integrated Circuits, McGraw-Hill, 3rd Ed 2004

1	**Modulbezeichnung** 92660	**Schaltungstechnik** Circuit technology	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Robert Weigel
5	**Inhalt**	• Halbleiterbauelemente: Diode, Bipolartransistor, MOSFET • Transistor-Grundschaltungen: Arbeitspunkte, Großsignal-, Kleinsignalverhalten • Verstärker: Stromquellen, Differenzverstärker, Impedanzwandler • Operationsverstärker, innerer Aufbau, Modelle, Anwendungen • Digital-Analog-/Analog-Digital-Umsetzer: Grundschaltungen, Modelle, Anwendungen
6	**Lernziele und Kompetenzen**	• Die Studierenden verstehen den Aufbau und die Funktionsweisen von Halbleiterschaltungen wie Dioden- und Transistorgrundschaltungen, Verstärkern, Operationsverstärkern und Analog-Digital-/Digital-Analog-Umsetzern und können diese erläutern. • Die Studierenden können komplexe Schaltungen durch eine Zerlegung in grundlegende Funktionsblöcke analysieren und diese in ihrer Funktion beurteilen. • Die Studierenden verstehen die Entwicklungsmethodik beim Entwurf von grundlegenden Halbleiterschaltungen und können diese dimensionieren. • Die Studierenden können eine einfache, abstrakte Funktionsbeschreibung in grundlegende Halbleiterschaltungen abbilden und diese zur Erfüllung der abstrakten Funktion auslegen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Mikroelektronik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Nebenfach EEI: Leistungselektronik

1	**Modulbezeichnung** 92590	**Halbleiterbauelemente** Semiconductor devices	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übungen zu Halbleiterbauelemente (2 SWS) Vorlesung: Halbleiterbauelemente (2 SWS) Tutorium: Tutorium Halbleiterbauelemente (2 SWS)	- 5 ECTS -
3	Lehrende	Jan Dick Prof. Dr.-Ing. Jörg Schulze	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Jörg Schulze
5	**Inhalt**	Das Modul Halbleiterbauelemente vermittelt den Studierenden der Elektrotechnik die physikalischen Grundlagen moderner Halbleiterbauelemente. Der erste Teil der Vorlesung befasst sich nach einer Einleitung in die moderne Halbleitertechnik und Halbleitertechnologie mit der Behandlung von Ladungsträgern in Metallen und Halbleitern; und es werden die wesentlichen elektronischen Eigenschaften der Festkörper zusammengefasst. Darauf aufbauend werden im Hauptteil der Vorlesung die Grundelemente aller Halbleiterbauelemente pn-Übergang, Schottky-Kontakt und MOS-Varaktor detailliert dargestellt. Damit werden dann zum Abschluss die beiden wichtigsten Transistorkonzepte der Bipolartransistor und der MOS-gesteuerte Feldeffekttransistor (MOSFET) ausführlich behandelt. Ein Ausblick, der die gesamte Welt der halbleiterbasierten Bauelemente für Logik- & Hochfrequenzanwendungen, Speicher- und leistungselektronischen Anwendungen beleuchtet, runden die Vorlesung ab.
6	**Lernziele und Kompetenzen**	Die Studierenden Fachkompetenz Verstehen • verstehen grundlegende physikalische Vorgänge (u.a. Drift, Diffusion, Generation, Rekombination) im Halbleiter • interpretieren Informationen aus Bänderdiagrammen Anwenden • beschreiben die Funktionsweisen moderner Halbleiterbauelemente • berechnen Kenngrößen der wichtigsten Bauelemente • übertragen - ausgehend von den wichtigesten Bauelementen, wie Dioden, Bipolartransistoren und Feldeffekttransistoren - diese Funktionsprinzipien auf Weiterentwicklungen für spezielle Anwendungsgebiete wie Leistungselektronik oder Optoelektronik Analysieren • diskutieren das Verhalten der Bauelemente z.B. bei hohen Spannungen oder erhöhter Temperatur
7	**Voraussetzungen für die Teilnahme**	Grundlagen der Elektrotechnik I
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!

9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Leistungselektronik Bachelor of Science Informatik 20222 Nebenfach EEI: Mikroelektronik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Vorlesungsskript, am LEB erhältlich • R. Müller: Grundlagen der Halbleiter-Elektronik, Band 1 der Reihe Halbleiter-Elektronik, Springer-Verlag, Berlin, 2002 • D.A. Neamen: Semiconductor Physics and Devices: Basic Principles, McGraw-Hill (Richard D. Irwin Inc.), 2002 • Th. Tille, D. Schmitt-Landsiedel: Mikroelektronik, Springer-Verlag, Berlin, 2004 • S.K. Banerjee, B.G. Streetman: Solid State Electronic Devices, Prentice Hall, 2005

1	**Modulbezeichnung** 96630	**Leistungselektronik** Power electronics	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übungen zu Leistungselektronik (2 SWS) Vorlesung: Leistungselektronik (2 SWS) Tutorium: Leistungselektronik Tutorium Kurs1 (0 SWS)	- 5 ECTS -
3	Lehrende	Prof. Dr. Martin März Nikolai Weitz Raffael Schwanninger Madlen Hoffmann Stefanie Büttner	

4	**Modulverantwortliche/r**	Thomas Eberle
5	**Inhalt**	*Grundlagen der Topologieanalyse*: Stationaritätsbedingungen, Strom-Spannungsformen, verbotene Schalthandlungen *Nicht-isolierende Gleichspannungswandler*: Grundlegende Schaltungstopologien, Funktionsweise, Dimensionierung *Isolierende Gleichspannungswandler*: Grundlegende Schaltungstopologien, Gleichrichterschaltungen, Transformatoren als Übertrager bzw. Energiespeicher *Leistungshalbleiter*: Grundlagen des statischen und dynamischen Verhaltens von MOSFET, IGBT und Dioden; Spezifika von WBG-Leistungshalbleitern auf Basis von Siliziumcarbid (SiC) und Galliumnitrid (GaN); Kommutierungsarten; Kurzschluss, Avalanche *Passive Leistungsbauelemente*: Induktive Bauelemente (weichmagnetische Kernmaterialien, nichtlineare Eigenschaften, Kernverluste, Wicklungsverluste); Kondensatoren (Technologien und deren Anwendungseigenschaften, sicherer Arbeitsbereich, Brauchbarkeitsdauer, Impedanzverhalten) *Parasitäre Elemente*: Niederinduktive Aufbautechniken *Treiber- und Ansteuerschaltungen für Leistungshalbleiter*: Grundschaltungen zur Ansteuerung MOS-gesteuerter Bauelemente mit und ohne galvanische Isolation, Schaltungen zur Erhöhung von Störabstand und Treiberleistung, Ladungspumpe, Schutzbeschaltungen, PWM-Modulatoren *Gleichrichter und Leistungsfaktorkorrektur*: Phasenan-/abschnittsteuerung, Netzstromverzerrungen, aktive Leistungsfaktorkorrektur, Gleichrichterschaltungen *Wechselrichter*: Netzgeführte Stromrichter, Zwei-/Dreipunktwechselrichter, Sinus-Dreieck- und Raumzeigermodulation
6	**Lernziele und Kompetenzen**	Die Studierenden können • die Funktionsprinzipien leistungselektronischer Basistopologien mit und ohne galvanische Isolation erklären, • einfache leistungselektronische Wandler analysieren und die für ein Systemdesign relevanten elektrischen und thermischen Parameter berechnen, • die grundlegenden Eigenschaften verschiedener Schaltungslösungen erklären und diskutieren,

		• die Vor- und Nachteile verschiedener Bauteiltechnologien in einer leistungselektronischen Schaltung bewerten, • einfache leistungselektronische Wandler entwerfen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Leistungselektronik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	[1] Franz Zach: Leistungselektronik. Springer-Vieweg, ISBN 978-3-658-04898-3 [2] Schröder D., Marquardt R.: Leistungselektronische Schaltungen. Springer-Vieweg, ISBN 978-3-662-55324-4 [3] Joachim Specovius: Grundkurs Leistungselektronik. Springer-Vieweg, ISBN 978-3-658-03308-8 [4] Ulrich Schlienz: Schaltnetzteile und ihre Peripherie. Vieweg, ISBN 3-528-03935-3 [5] Albach M.: Induktivitäten in der Leistungselektronik. Springer-Vieweg, ISBN 978-3-658-15080-8 [6] Tursky W., Reimann T., et al.: Applikationshandbuch Leistungshalbleiter. Semikron, ISBN 978-3-938843-56-7 [7] Volke A., Hornkamp M.: IGBT Modules. Infineon, ISBN 978-3-00-040134-3 [8] Kenneth L. Kaiser: Electromagnetic Compatibility Handbook. CRC Press, ISBN 0-8493-2087-9 [9] Hofer K.: Moderne Leistungselektronik und Antriebe. VDE-Verlag, ISBN 3-8007-2067-1

1	**Modulbezeichnung** 92650	**Regelungstechnik A (Grundlagen)** Control engineering A (Foundations)	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Regelungstechnik A (Grundlagen) (4 SWS)	5 ECTS
3	Lehrende	Prof. Dr.-Ing. Knut Graichen	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Knut Graichen
5	**Inhalt**	Die Vorlesung behandelt die Grundlagen der Regelungstechnik und befähigt zur Beschreibung und Untersuchung linearer Systeme und zum Entwurf einfacher und mehrschleifiger Regler im Frequenzbereich. Die Inhalte der Vorlesung sind: • Gegenstand und Zielstellung der Regelungstechnik • Modellbildung der Strecke im Zeit und Frequenzbereich und Darstellung als Strukturbild • Analyse des Streckenverhaltens linearer Eingrößensysteme anhand von Übertragungsfunktion und Frequenzgang • Auslegung einschleifiger Regelkreise • Erweitere Regelkreisstrukturen
6	**Lernziele und Kompetenzen**	Die Studierenden können • Gegenstand und Zielstellung der Regelungstechnik erläutern. • Problemstellungen als Steuerungs- und Regelungsaufgabe identifizieren. • das Streckenverhalten durch ein mathematisches Modell in Form des Strukturbilds beschreiben. • eine Modellvereinfachung durch Linearisierung und Strukturbildumformung durchführen. • aus Übertragungsfunktion und Frequenzgang das qualitative Streckenverhalten ermitteln. • zu einem Frequenzgang Ortskurve und Bode-Diagramm angeben. • den Aufbau einer Zwei-Freiheitsgrade-Regelung angeben und die Zweckbestimmung von Vorsteuerung und Regelung erläutern. • Sollverläufe auf Zulässigkeit überprüfen und realisierbare Vorsteuerungen entwerfen. • die Regelkreis-Stabilität definieren und mit dem Nyquist-Kriterium untersuchen. • entscheiden, wann welcher Reglertyp in Frage kommt und nach welchen Gesichtspunkten dessen Parameter zu wählen sind. • für lineare Eingrößensysteme einen geeigneten Regler entwerfen. • ergänzende Maßnahmen zur Störverhaltensverbesserung beschreiben und zur Anwendung bringen. • die Vorlesungsinhalte auf verwandte Problemstellungen übertragen und sich weiterführende Frequenzbereichsmethoden der Regelungstechnik selbständig erschließen.

7	**Voraussetzungen für die Teilnahme**	Systemtheorie linearer zeitkontinuierlicher Systeme (inkl. Laplace-Transformation)
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach EEI: Automatisierungstechnik Bachelor of Science Informatik 20222 Nebenfach EEI: Leistungselektronik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• O. Föllinger. Regelungstechnik: Einführung in die Methoden und ihre Anwendung, 12. Auflage, VDE-Verlag, 2016 • M. Horn, N. Dourdoumas. Regelungstechnik, Pearson Studium, 2004 • W. Leonhard. Einführung in die Regelungtechnik, 4. Auflage, Vieweg, 1987 • J. Lunze. Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen, 12. Auflage, Springer, 2020 • R. Unbehauen. Regelungtechnik 1, 12. Auflage, 2002 • G. Ludyk. Theoretische Regelungstechnik 1 und 2, Springer, 1995

Nebenfach Maschinenbau: Grundlagen der Produktentwicklung

1	**Modulbezeichnung** 94711	**Grundlagen der Produktentwicklung** Foundations of product development	**7,5 ECTS**
2	Lehrveranstaltungen	Übung: Maschinenelemente Übung (2 SWS) Vorlesung: Vorlesung Maschinenelemente I (4 SWS) Tutorium: Tutorium Maschinenelemente I (2 SWS)	- - -
3	Lehrende	Dr.-Ing. Marcel Bartz Michael Jüttner	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Sandro Wartzack
5	**Inhalt**	**Einführung in die Produktentwicklung** • Synthese und Analyse als zentrale Aufgaben • Vorgehensmodelle im Produktentwicklungsprozess **Konstruktionswerkstoffe** **Grundlagen der Bauteilauslegung Festigkeitslehre** • Typische Versagenskriterien • Definition und Aufgaben der Festigkeitslehre, Prinzip • Ermittlung von Belastungen • Ermittlung von Beanspruchungen • Beanspruchungsarten • Zeitlicher Verlauf der Beanspruchung und Lastannahmen • Resultierende Spannungen und Vergleichsspannungen • Kerbwirkung und Stützwirkung • Weitere Einflussfaktoren auf die Festigkeit von Bauteilen • Maßgebliche Werkstoffkennwerte • Bauteildimensionierung und Festigkeitsnachweis **Einführung in die Technische Produktgestaltung** • Gestalten von Maschinen • Fertigungsgerechtes Gestalten • Sicherheitsgerechtes Gestalten **Normung, Toleranzen, Passungen und Oberflächen** **Maschinenelemente** • Schweißverbindungen • Passfeder- und Keilwellenverbindungen • Bolzen- und Stiftverbindungen • Zylindrische Pressverbindungen • Kegelverbindungen • Spannelementverbindungen • Schraubenverbindungen • Wälzlager • Gleitlager • Dichtungen • Stirnräder und Stirnradgetriebe • Kupplungen
6	**Lernziele und Kompetenzen**	**<u>Fachkompetenz</u>** **Wissen** Im Rahmen von GPE erlangen die Studierenden grundlegende Kenntnisse im Bereich der Maschinenelemente. Die Studierenden

sind vertraut mit Fachbegriffen und können Wissen zu folgenden Themenbereichen wiedergeben:

- Gestalten von Maschinenbauteilen unter besonderer Berücksichtigung der Fertigungsgerechtheit
- Normen (DIN, EN, ISO), Richtlinien (VDI, FKM) und Standards im Kontext des Maschinenbaus
- herstell- und messbedingte Abweichungen sowie zu vergebende Toleranzen für Maß, Form, Lage und Oberfläche bei Maschinenbauteilen
- rotatorische Wälzlager und Wälzlagerungen, insbesondere Wissen über die gängigen Radial- und Axialwälzlagerbauformen, deren spezifische Merkmale und Eigenschaften sowie deren sachgerechte Einbindung in die Umgebungskonstruktion
- Getriebe als wichtige mechanische Komponente in Antriebssträngen

Verstehen

Die Studierenden verstehen Zusammenhänge zu erarbeitetem Wissen durch die Erschließung von Querverbindungen zu den in folgenden Lehrveranstaltungen erworbenen bzw. zu erwerbenden Kompetenzen:

- Lehrveranstaltung Produktionstechnik und Technische Produktgestaltung
- Lehrveranstaltung Technische Darstellungslehre
- Lehrveranstaltung Messtechnik

Die Studierenden gewinnen ein allgemeines Verständnis für:

- das Konstruieren von Maschinen als methodischer Prozess unter besonderer Beachtung von Synthese und Analyse als zentrale Aufgaben der Produktentwicklung und auf Basis der Begriffe Merkmale und Eigenschaften nach der Definition von WEBER. Mit Fokus auf VDI 2221 ff verstehen die Studierenden Vorgehensmodelle in Produktentwicklungsprozessen. Hierbei werden Querverweise zu den in der Lehrveranstaltung Methodisches und rechnerunterstütztes Konstruieren zu erwerbenden Kompetenzen aufgezeigt.
- die Konstruktionswerkstoffe, deren spezifische Eigenschaften sowie Möglichkeiten zur Beschreibung des Festigkeits-, Verformungs- und Bruchverhaltens. Unter Konstruktionswerkstoffen werden insbesondere Eisenwerkstoffe, daneben auch Nichteisenmetalle, Polymerwerkstoffe und spezielle neue Werkstoffe, z. B. Verbundwerkstoffe, verstanden. Es werden Querverbindungen zu den in der Lehrveranstaltung Werkstoffkunde erworbenen Kompetenzen erschlossen.

Die Studierenden gewinnen ein Verständnis für Maschinenbauteile im Hinblick auf deren rechnerische Auslegung und konstruktive Gestaltung unter Berücksichtigung des Werkstoffverhaltens, der Geometrie und der auf das Bauteil einwirkenden Lasten. Hierzu:

- Unterscheidung von Nennspannungen und örtlichen Spannungen
- Verständnis für mehrachsige Beanspruchungszustände und Festigkeitshypothesen in Verbindung mit den werkstoffspezifischen Versagenskriterien
- Verständnis für die Auswirkungen von Kerben auf Maschinenbauteile unter statischer und dynamischer Beanspruchung
- Verständnis für Werkstoffkennwerte und den Einfluss der Bauteilgröße und des Oberflächenzustandes sowie Gegenüberstellung zu dazugehörigen Versagenskriterien.

Die Studierenden gewinnen ein funktionsorientiertes Verständnis für und Überblick zu gängigen Maschinenelementen sowie Vertiefung zahlreicher Maschinenelemente unter Berücksichtigung derer spezifischen Merkmale, Eigenschaften und Einsatzbedingungen. Insbesondere wird hierbei ein Schwerpunkt auf das Erlangen eines Verständnisses für Wirkprinzipien und Gestaltung gelegt. Im Einzelnen für:

- Schweißverbindungen
- formschlüssige Welle-Nabe-Verbindungen
- Bolzen- und Stiftverbindungen
- reibschlüssige Welle-Nabe-Verbindungen
- Elemente von Schraubenverbindungen unter besonderer Berücksichtigung des Maschinenelements Schraube (Gewinde), sowie Schaubensicherungen
- rotatorische Wälzlager und Wälzlagerungen. Hierzu ein Verständnis für die konstruktive Gestaltung von Wälzlagerstellen, insbesondere Passungswahl und Lageranordnungen
- statische und dynamische Dichtungen und deren Klassifizierung sowie die Auswahl von Dichtungen unter Berücksichtigung gegebener technischer Randbedingungen
- Basiswissen über Antriebssysteme, Antriebsstränge und Antriebskomponenten, Verständnis für Last- und Beschleunigungsdrehmomente und zu reduzierende Trägheitsmomente. Hierbei Aufzeigen von Querverweisen zu den in den Lehrveranstaltungen Regelungstechnik und Elektrische Antriebstechnik zu erwerbenden Kompetenzen
- Zahnradgetriebe mit Fokus auf Stirnräder und Stirnradgetriebe. Hierbei Verständnis des Verzahnungsgesetzes und der Geometrie der Evolventenverzahnung für Gerad- und Schrägverzahnung
- nicht-schaltbare und schaltbare Kupplungen, Klassifizieren von Kupplungen nach deren Funktions- und Wirkprinzipien, Auswahl von Kupplungen unter Berücksichtigung gegebener technischer Randbedingungen

Anwenden

Die Studierenden vertiefen Teile des unter Punkt 1.2 beschriebenen Verständnisses durch die Anwendung von spezifischen

Berechnungsmethoden. Dies umfasst insbesondere folgende Themenbereiche:

- Berechnung von Maßtoleranzen
- Berechnung von Schweißverbindungen und der Tragfähigkeit von Schweißverbindungen nach dem Verfahren von NIEMANN
- Berechnung formschlüssiger Welle-Nabe-Verbindungen, insbesondere Passfederverbindungen auf Basis von DIN 6892 und Keilwellenverbindungen sowie deren Gültigkeitsgrenzen
- Berechnung einfacher Bolzen- und Stiftverbindungen sowie deren Gültigkeitsgrenzen
- Berechnung von zylindrischen Quer- und Längspressverbänden in Anlehnung an DIN 7190 (elastische Auslegung) sowie von Kegelpressverbänden
- Überprüfung längs- und querbelasteter, vorgespannter Schraubenverbindungen in Anlehnung an VDI 2230 im Hinblick auf Anziehdrehmoment, Bruch, Fließen und Dauerbruch der Schraube unter Einfluss von Setzvorgängen und Schwankungen beim Anziehen
- Berechnung der Tragfähigkeit von Wälzlagern für statische und dynamische Betriebszustände auf Basis von DIN ISO 76 und DIN ISO 281 (nominelle und erweiterte modifizierte Lebensdauer
- Berechnung von Übersetzungen, Wirkungsgraden und Drehmomentverhältnissen in Getrieben
- Berechnung von Verzahnungsgeometrien auf Basis von DIN 3960
- Berechnung von am Zahnrad wirkenden Kräften und Ermittlung der Zahnfuß- und der Grübchentragfähigkeit in Anlehnung an DIN 3990 sowie deren Gültigkeitsgrenzen

Analysieren

Die Studierenden erlernen mithilfe dem Verständnis aus 1.2 und den Berechnungsmethoden aus 1.3 definierte Problemstellungen im Kontext der Maschinenelemente sowie deren Zusammenwirken zu lösen. Hierzu gehört:

- Analyse der auf ein Bauteil wirkenden Belastungen. Hierbei erschließen von Querverbindungen zu den in der Lehrveranstaltung Statik erworbenen Kompetenzen
- Analyse der aus den Belastungen resultierenden Beanspruchungen mit Fokus auf die Beanspruchung stabförmiger Bauteile, Kontaktbeanspruchung sowie Instabilität stabförmiger Bauteile (Knicken). Hierbei erschließen von Querverbindungen zu den in der Lehrveranstaltung Elastostatik erworbenen Kompetenzen
- Analyse und Beurteilung von Lastannahmen sowie des zeitlichen Verlaufs von Beanspruchungen (statisch, dynamisch)
- Ermittlung von Kerbspannungen auf Basis von Kerbform-, Kerbwirkungszahlen und plastischen Stützzahlen unter Berücksichtigung von Oberflächeneinflüssen

		• Auswahl von Vergleichsspannungshypothesen und Ermittlung von Vergleichsspannungen • Auswahl von Maßtoleranzen • Auswahl von Wälzlagern und Grobgestaltung von Wälzlagerstellen. Hierbei erschließen von Querverbindungen zu den in der Lehrveranstaltung Wälzlagertechnik zu erwerbenden Kompetenzen • Auswahl gängiger Maschinenelemente unter Funktionsgesichtspunkten sowie Auslegen ausgewählter Maschinenelemente **Evaluieren (Beurteilen)** Die Studierenden erlernen über die Analyse hinaus die Möglichkeiten zur Einschätzung ihrer Berechnungen. Besonderer Schwerpunkt liegt hierbei auf der Überprüfung der Festigkeit von Maschinenbauteilen im Zuge von Dimensionierungsaufgaben und Tragfähigkeitsnachweisen in Anlehnung an die einschlägige FKM-Richtlinie sowie Beurteilung der durchgeführten Berechnungen unter besonderer Berücksichtigung von Unsicherheiten, welche Ausdruck in der Wahl von Mindestsicherheiten finden. Die Studierenden erlernen somit Möglichkeiten zur Beurteilung von: • Auswahl und Auslegung von Maschinenelementen unter Funktionsgesichtspunkten • Auswahl und Auslegung von Maschinenelementen unter Tragfähigkeitsgesichtspunkten **<u>Lern- bzw. Methodenkompetenz</u>** Die Studierenden erlernen Verfahren und Methoden zur Einschätzung und Bewertung von Maschinenelementen, einschließlich der Befähigung, Berechnungsansätze und Gestaltungsgrundsätze auch auf andere Maschinenelemente, die nicht explizit im Rahmen der Lehrveranstaltung behandelt wurden, zu übertragen.
7	**Voraussetzungen für die Teilnahme**	Es werden empfohlen: • Technische Darstellungslehre I • Statik und Festigkeitslehre
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Grundlagen der Produktentwicklung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 120 h Eigenstudium: 105 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 94591	**Technische Darstellungslehre 2** Engineering drawing	**2,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Sandro Wartzack
5	**Inhalt**	• Technologie des Computer Aided Design • Einführung in die virtuelle Produktentwicklung mit CAD-Systemen • Grundlagen des CAD: Arten von 3D-Modellierern, Systemmodule und Eigenschaften von Modellen • Modellierungsstrategien, Vorgehensweise bei der Modellierung, Grundprinzipien, Besondere Modellierungsvereinfachungen im Zusammenhang mit genormten Darstellungen • Rechnerübung mit Hausübung an CAD-Systemen zum Anfertigen von Bauteilen, Baugruppen und technischen Zeichnungen
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden erwerben Wissen über Funktion, Aufbau und Bedienung von im industriellen Umfeld eingesetzten, vollparametrischen 3D-CAD-Systemen. Die Studierenden erfahren die Bedeutung von CAD-Systemen als zentralem Synthesewerkzeug des rechnerunterstützten Produktentwicklungsprozesses im Maschinenbau und in verwandten Disziplinen. Hierzu erlernen die Studierenden Grundwissen über die einzelnen Phasen des Produktlebenszyklus und die Möglichkeiten der Rechnerunterstützung (CAx) und über die grundlegende Geometrieverarbeitung auf Rechnersystemen: Historische Entwicklung, Stand der Technik, Grundfunktionalitäten moderner CAD-Systeme, Parametrik, Assoziative Datenspeicherung, Features und Konstruktionselemente, historienbasierte und direkte Modellierung. Verstehen Die Studierenden gewinnen Verständnis für den Einsatz von CAD zur Definition der Produktgestalt im Hinblick auf eine durchgängige Verwendung der erzeugten Daten als Grundlage für die weitere CAx-Werkzeuge sowie für die Ableitung normgerechter Zeichnungen und Stücklisten. Weiterhin vertiefen die Studierenden ihr Verständnis für technische Zeichnungen gemäß DIN 199-1 indem sie die technische Zeichnung sowohl als Informationsquelle für die Modellerstellung verwenden, als auch normgerechte technische Zeichnungen ihrer erstellten 3D-CAD Modelle erstellen. Durch diese doppelte Verwendung des Kommunikationsmediums technische Zeichnung erwerben die Studierenden ein Verständnis für den Zusammenhang zwischen der technischen Zeichnung und dem 3D-CAD Modell.

Die Studierenden verstehen Zusammenhänge durch:

- Aufzeigen von Querverweisen zu den im Vorpraktikum (verpflichtend für MB und WING) erworbenen Kompetenzen
- Aufzeigen von Querverweisen zu den in der Lehrveranstaltung Technische Darstellungslehre I erworbenen Kompetenzen

Anwenden
Die Studierenden wenden das Gelernte an, indem sie das in den obligatorischen Grundlagenübungen vermittelte Wissen in verpflichtenden Vertiefungsübungen auf bisher unbekannte Bauteile transferieren. Die Studierenden werden bei der Anwendung des Wissens durch die Korrektur der abgegebenen 3D-Modelle und Zeichnungen überprüft. Den Studierenden werden die gemachten Fehler erklärt und die Möglichkeit gegeben diese zu korrigieren und so das Wissen zur Erstellung korrekter 3D-Modelle und Zeichnungen anzuwenden.
Analysieren
Die Studierenden vertiefen die in der Lehrveranstaltung Technische Darstellungslehre I erlangten Analysefähigkeiten technischer Zeichnungen und Funktionsskizzen zur anschließenden Erstellung eines korrekten CAD-Modells auf Basis der relevanten Informationen.
Evaluieren (Beurteilen)
Die Studierenden lesen selbstständig technische Zeichnungen und Funktionsskizzen und erlernen, deren Inhalt zu erfassen, zu beurteilen und zu hinterfragen. Dies beinhaltet insbesondere die Entwicklung einer korrekten Modellierungsstrategie auf Basis der erfassten Informationen.
Die Studierenden bewerten im Rahmen der Baugruppenkonstruktionen selbstständig die funktionsrelevanten Kontakt- und Funktionsflächen der Einzelbauteile.
Erschaffen
Die Studierenden erstellen Einzelteile durch Modellieren von Volumenkörpern unter Berücksichtigung einer robusten Modellierungsstrategie, hierzu zählen:
Definieren von Geometriereferenzen und zweidimensionalen Skizzen als Grundlage für Konstruktionselemente
Erzeugen von Volumenkörpern mit Hilfe der Konstruktionselemente Profilextrusion, Rotation, Zug und Verbund
Kombinieren von Volumenkörpern durch boolesche Operationen zu Rohbauteilen gemäß eines spannenden Fertigungsverfahrens
Detaillieren von Rohbauteilen durch Hinzufügen von Bohrungen, Fasen und Metainformationen (z. B. Toleranzangaben)
Nachträgliches Ändern der Geometrie mit Hilfe von Parametrik.
Die Studierenden erstellen Baugruppen durch Kombination von Einzelteilen unter Verwendung von Normteilbibliotheken, hierbei erlangen die Studierenden insbesondere Kompetenzen in den Bereichen:
Planen einer Baugruppenhierarchie im Hinblick auf Robustheit
Verarbeiten von Importgeometrie (Fremdformate)
Definieren von Montagebedingungen

		Anwenden einfacher Baugruppenanalysefunktionen (z. B. Durchdringung und Masseeigenschaften). Die Studierenden erlernen das Ableiten norm-, funktions- und fertigungsgerechter Einzelteil- und Zusammenbauzeichnungen aus den 3D-CAD-Modellen, welche den Regeln der Technischen Darstellungslehre folgen, hierbei vertiefen die Studierenden die in der Lehrveranstaltung Technische Darstellungslehre I erworbenen Kompetenzen. Die Studierenden erwerben die Befähigung zum Erstellen auch komplexerer Einzelteile und Baugruppen in 3D-CAD-Systemen und zum Ableiten zugehöriger technischer Zeichnungen sowie Befähigung, sich Modellierungsmöglichkeiten zu erschließen, die nicht explizit im Rahmen der Lehrveranstaltung behandelt wurden und Befähigung, die gewonnenen Erkenntnisse auf andere als im Rahmen der Lehrveranstaltung eingesetzte 3D-CAD-Systeme übertragen zu können. Lern- bzw. Methodenkompetenz Die Studierenden erlangen die zuvor genannten Fachkompetenzen insbesondere durch praktische Anwendung des 3D-CAD-Systems im Rahmen verpflichtender Übungsabgaben. Individuelle und kompetente Betreuung erhalten die Studierenden hierbei durch studentische Tutoren und Mitarbeiter des Lehrstuhls, wodurch sichergestellt wird, dass eine effiziente Vermittlung der Lehrinhalte trotz unterschiedlichem Kenntnisstandes der Studierenden erfolgt. Selbstkompetenz Die Studierenden vertiefen die Befähigung zur selbstständige Arbeitseinteilung und Einhaltung von Meilensteinen bei der Bearbeitung der verpflichtenden Aufgaben, hierbei erhalten die Studierenden Unterstützung durch Betreuer und studentische Tutoren. Die Studierenden reflektieren die eigenen Stärken und Schwächen beim Umgang mit dem 3D-CAD-System und fordern eigenverantwortlich Hilfe bei den studentischen Tutoren und Mitarbeitern des Lehrstuhls ein und erweitern ihre Selbstkompetenz bezüglich des Umgangs mit fachlicher Kritik. Sozialkompetenz Die Studierenden vertiefen die Befähigung zur selbstständige Arbeitseinteilung und Einhaltung von Meilensteinen bei der Bearbeitung der verpflichtenden Aufgaben, hierbei erlangen sie die Befähigung zur kooperativen Zusammenarbeit mit anderen Studierenden, studentischen Tutoren und Mitarbeitern des Lehrstuhls.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Grundlagen der Produktentwicklung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung

11	**Berechnung der Modulnote**	Praktikumsleistung (0%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 45 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 97110	**Technische Produktgestaltung** Technical product design	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Sandro Wartzack
5	**Inhalt**	• Einführung in die Technische Produktgestaltung • Baustrukturen technischer Produkte • Fertigungsgerechte Werkstückgestaltung • toleranzgerechtes Konstruieren • kostengerechtes Konstruieren • beanspruchungsgerechtes Konstruieren • werkstoffgerechtes Konstruieren • Leichtbau • umweltgerechtes Konstruieren • nutzerzentrierte Produktgestaltung
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Im Rahmen von TPG erwerben die Studierenden Kenntnisse zur Berücksichtigung verschiedener Aspekte des Design-for-X bei der Entwicklung technischer Produkte. Nach der erfolgreichen Teilnahme kennen sie die jeweiligen Gestaltungsrichtlinien und zugehörige Methoden. Dies sind im Einzelnen: • Wissen über Möglichkeiten zur Umsetzung des Leichtbaus und daraus abgeleitet über spezifische Gestaltungsrichtlinien, die im Rahmen des Leichtbaus zu berücksichtigen sind, hierzu: Beanspruchungsgerechtes Konstruieren (Kraftfluss, Prinzip der konstanten Gestaltfestigkeit, Kerbwirkung, Prinzip der abgestimmten Verformung, Prinzip des Kräfteausgleichs) • Wissen über werkstoffgerechtes Konstruieren (Anforderungs- und Eigenschaftsprofil, wirtschaftliche Werkstoffauswahl, Auswirkung der Werkstoffwahl auf Fertigung, Lebensdauer und Gewicht) • Wissen über die Auswirkungen eines Produktes (und insbesondere der vorhergehenden Konstruktion) auf Umwelt, Kosten und den Nutzer, hierzu: Umweltgerechtes Konstruieren (Recycling, Einflussmöglichkeiten in der Produktentwicklung, Strategien zur Berücksichtigung von Umweltaspekten, Life Cycle Assessment, Produktinstandsetzung, Design for Recycling) • Wissen über kostengerechtes Konstruieren (Beeinflussung der Lebenslauf-, Herstell- und Selbstkosten in der Produktentwicklung, Auswirkungen der Stückzahl und der Fertigungsverfahren, Entwicklungsbegleitende Kalkulation) • Wissen über nutzerzentrierte Produktentwicklung (Anthropometrie, Nutzerintegration in der Produktentwicklung, Mensch-Maschine-Schnittstellen, Beeinträchtigungen im Alter,

Universal Design, Gestaltungsrichtlinien nach dem SENSI-Regelkatalog, etc.)

- Wissen über spezifische Gestaltungsrichtlinien der Fertigungsverfahren des Urformens" (Gießen, Pulvermetallurgie, Additive Fertigung)
- Wissen über spezifische Gestaltungsrichtlinien der Fertigungsverfahren des Umformens" (Schmieden, Walzen, Biegen, Scheiden, Tiefziehen, Stanzen, Fließpressen)
- Wissen über spezifische Gestaltungsrichtlinien der Fertigungsverfahren des Trennens" (Zerteilen, Drehen, Fräsen, Bohren, Schleifen, Erodieren)
- Wissen über spezifische Gestaltungsrichtlinien der Fertigungsverfahren des Fügens" (Schweißen, Löten, Nieten, Durchsetzfügen, Kleben, Fügen durch Urformen)
- Wissen über spezifische Gestaltungsrichtlinien der Fertigungsverfahren des Beschichtens und Stoffeigenschaften ändern" (Schmelztauchen, Lackieren, Thermisches Spritzen, Physical Vapour Deposition, Chemical Vapour Deposition, Galvanische Verfahren, Pulverbeschichten, Vergüten, Glühen)
- Wissen über spezifische Gestaltungsrichtlinien des montagegerechten Konstruierens bzgl. der Baustruktur technischer Produkte (Integral-, Differential und Verbundbauweise, Produktstrukturierung, Variantenmanagement, Modularisierung) und des Montageprozesses (Gestaltung der Fügeteile und Fügestellen, Automatisches Handhaben und Speichern, Toleranzausgleich, DFMA)
- Wissen über spezifische Inhalte des toleranzgerechten Konstruierens (insbesondere Grundlage der geometrischen Tolerierung und die Vorgehensweise zur Vergabe von Toleranzen)

Verstehen

Nach erfolgreichem Abschluss des Moduls Technische Produktgestaltung" verfügen die Studierenden über Verständnisse hinsichtlich der technischen und nicht-technischen Einflussfaktoren und deren Abhängigkeiten bei der Gestaltung technischer Produkte ausgehend von der Produktstruktur bis zur konstruktiven Bauteilgestaltung. Hierbei stehen besonders die folgenden Verständnisse im Fokus:

- Verständnis über die Spezifikation von Toleranzen, Passungen und Oberflächen in Technischen Zeichnungen unter Berücksichtigung deren Auswirkungen auf Fertigung, Montage und den Betrieb des Produktes, hierzu: Verständnis der Vorgehensweise zur Toleranzspezifikation sowie erforderlicher Grundlagen zur Tolerierung von Bauteilen (Allgemeintoleranzen, wirkliche und abgeleitete Geometrieelemente, Hüllbedingung, Unabhängigkeitsprinzip, Inklusion verschiedener Toleranzarten, Bezugssysteme und Ausrichtungskonzepte, statistische Toleranzanalyse, etc.)

- Verständnis über Fertigung und Montage sowie über die Bedeutung des Design-for-X und insbesondere des fertigungsgerechten Konstruierens im Produktentwicklungsprozess
- Verständnis über die Berücksichtigung nicht-technischer Faktoren, wie beispielsweise Umwelt-, Kosten- und Nutzeraspekten, und deren Wechselwirkungen bei der Gestaltung technischer Produkte.

Anwenden

Die Studierenden wenden im Rahmen von Übungsaufgaben Gelerntes an. Dabei werden bestehende Entwürfe und Konstruktionen durch die Studierenden entsprechend der vermittelten Gestaltungsrichtlinien optimiert und neue Konstruktionen unter Einhaltung dieser Gestaltungsrichtlinien erschaffen. Dies beinhaltet im Einzelnen:

- Erstellung der fertigungsgerechten und montagegerechten Tolerierung von Bauteilen. Dies umschließt folgende Tätigkeiten: Bestimmen der zugrundeliegenden Bezugssysteme und Ausrichtungskonzepte; Bestimmen des Tolerierungsgrundsatzes. Integration von, durch Normen definierte Toleranz- und Passungsvorgaben in bestehende Tolerierungen; Zusammenfassen kombinierbarer Form- und Lagetoleranzen zu Zeichnungsvereinfachung; Festlegung der Größen der Toleranzzonen aller vergebenen Toleranzen.
- Optimierung der Tolerierung anhand der statistischen Toleranzanalyse. Dies umschließt folgende Tätigkeiten: Erkennen und Ableiten der analytischen Schließmaßgleichungen; Definition der zugrundeliegenden Toleranzwerten und zugehörigen Wahrscheinlichkeitsverteilungen; Berechnung der resultierenden Wahrscheinlichkeitsverteilungen der Schließmaße; analytische Bestimmung der statistischen Beitragsleister mittels lokaler Sensitivitätsanalysen; Beurteilung der Ergebnisse und ggf. anschließende Anpassung der Tolerierung der Bauteile; Transfer der Ergebnisse auf zeitabhängige Mechanismen (kinematische Systeme).
- Änderung der Gestaltung von Bauteilen, bedingt durch die Änderung der zu fertigenden Stückzahl der Baugruppe. Dies umschließt die folgenden Tätigkeiten: Bestimmung des konstruktiven Handlungsbedarfs; Anpassung der Gestaltung der Bauteile insbesondere hinsichtlich der fertigungsgerechten und der montagegerechten Gestaltung. Gestaltung der erforderlichen Werkzeuge zur Fertigung der Bauteile und Bewertung dieser bzgl. der resultierenden Kosten.

Analysieren

- Aufzeigen von Querverweisen zu den im Modul Produktionstechnik zu erwerbenden Kompetenzen über die Hauptgruppen der Fertigungsverfahren nach DIN 8580

		• Aufzeigen von Querverweisen zu den im Modul Handhabungs- und Montagetechnik zu erwerbenden Kompetenzen über montagegerechtes Konstruieren • Aufzeigen von Querverweisen zu den im Modul Umformtechnik zu erwerbenden Kompetenzen über Fertigungsverfahren der Hauptgruppe Umformen nach DIN 8580 Evaluieren (Beurteilen) Anhand der erlernten Grundlagen über unterschiedliche Aspekte des Design-for-X, deren Berücksichtigung bei der Gestaltung technischer Produkte durch Gestaltungsrichtlinien, Methoden, und Vorgehensweisen sowie den dargelegten Möglichkeiten zur Rechnerunterstützung können die Studierenden kontextbezogene Richtlinien für die Gestaltung technischer Produkte in unbekannten Konstruktionsaufgaben auswählen und deren Anwendbarkeit einschätzen. Zudem sind sie in der Lage konträre Gestaltungsrichtlinien aufgabenspezifisch abzuwägen. Erschaffen Die Studierenden werden durch die erlernten Grundlagen befähigt, konkrete Verbesserungsvorschläge zu bestehenden Konstruktionen hinsichtlich unterschiedlicher Design-for-X Aspekte eigenständig zu erarbeiten. Zudem sind sie in der Lage technische Produkte so zu gestalten, dass diese verschiedenste technische und nicht-technische Anforderungen (fertigungsbezogene Anforderungen, Kostenanforderungen, Umweltanforderungen, Nutzeranforderungen, etc.) bedienen. Darüber hinaus werden die Studierenden in die Lage versetzt, Gestaltungsrichtlinien für neuartige Fertigungsverfahren aus grundlegenden Verfahrenseigenschaften abzuleiten und bei der Gestaltung technischer Produkte anzuwenden. Lern- bzw. Methodenkompetenz Befähigung zur selbständigen Gestaltung von Produkten und Prozessen gemäß erlernter Vorgehensweisen und Richtlinien sowie unter verschiedensten Design-for-X-Aspekten sowie zur objektiven Bewertung bestehender Produkte und Prozesse hinsichtlich gestellter Anforderungen des Design-for-X. Selbstkompetenz Befähigung zur selbständigen Arbeitseinteilung und Einhaltung von Meilensteinen. Objektive Beurteilung sowie Reflexion der eigenen Stärken und Schwächen sowohl in fachlicher (u. a. Umsetzung der gelehrten Richtlinien des Design-for-X in der Konstruktion) als auch in sozialer Hinsicht (u. a. Erarbeitung von Lösungen und Kompromissen im interdisziplinären Team). Sozialkompetenz Die Studierenden organisieren selbstständig die Bearbeitung von Übungsaufgaben in kleinen Gruppen und erarbeiten gemeinsam Lösungsvorschläge für die gestellten Übungsaufgaben. In der gemeinsamen Diskussion erarbeiteter Lösungen geben Betreuer und Kommilitonen wertschätzendes Feedback.
7	**Voraussetzungen für die Teilnahme**	Keine

8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Grundlagen der Produktentwicklung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Nebenfach Maschinenbau: Fertigungsautomatisierung und Produktionstechnik

1	**Modulbezeichnung** 94570	**Produktionstechnik I und II** Production engineering I+II	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Produktionstechnik II (2 SWS) Vorlesung: Produktionstechnik I (2 SWS) Tutorium: Produktionstechnik II - Tutorium (2 SWS, SoSe 2024) Tutorium: Produktionstechnik I Tutorium (SWS)	2,5 ECTS 2,5 ECTS 2,5 ECTS -
3	Lehrende	Prof. Dr. Nico Hanenkamp Prof. Dr.-Ing. Dietmar Drummer Prof. Dr.-Ing. Jörg Franke Prof. Dr.-Ing. Marion Merklein Prof. Dr.-Ing. Michael Schmidt Simon Sauer Prof. Dr. Hinnerk Hagenah	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Marion Merklein
5	**Inhalt**	*Produktionstechnik I:* Basierend auf der DIN 8580 werden die aktuellen Technologien sowie die dabei eingesetzten Maschinen in den Bereichen Urformen, Umformen, Trennen, Fügen, Beschichten und das Ändern der Stoffeigenschaften behandelt. Hierbei werden sowohl die Prozessketten als auch die spezifischen Eigenschaften der Produktionstechniken aufgezeigt und anhand von praxisrelevanten Bauteilen erläutert. Zum besseren Verständnis der Verfahren werden zunächst metallkundliche Grundlagen, wie der mikrostrukturelle Aufbau von metallischen Werkstoffen und ihr plastisches Verhalten, erläutert. Anschließend werden die Urformverfahren Gießen und Pulvermetallurgie dargestellt. Im weiteren Verlauf erfolgt eine Gegenüberstellung der Verfahren der Massivumformung Stauchen, Schmieden, Fließpressen und Walzen. Im Rahmen des Kapitels Blechumformung wird die Herstellung von Bauteilen durch Tiefziehen, Streckziehen und Biegen betrachtet. Der Fokus in der Vorstellung der Verfahrensgruppe Trennen liegt auf den Prozessen des Zerteilens und Spanens. Der Bereich Fügen behandelt die Herstellung von Verbindungen mittels Umformen, Schweißen und Löten. Abschließend werden verschiedene strahlbasierte Fertigungsverfahren aus den sechs Bereichen vorgestellt. Im Fokus stehen hierbei laserbasierte Fertigungsverfahren, wie zum Beispiel Schweißen, Schneiden oder Additiven Fertigung. Eine zusätzlich angebotene Übung dient der Vertiefung und der Anwendung des Vorlesungsinhaltes. *Produktionstechnik II:* Es wird die Verarbeitung von Kunststoffen (Spritzgießen, Erzeugung von duroplastischen / thermoplastischen Faserverbunden) und Metallen mit dem Fokus auf strahlbasierten Verfahren (Schneiden, Schweißen und Additive Fertigung mittels Wasser-, Elektronen- und Laserstrahl)behandelt. Des Weiteren werden die Grundlagen zu Werkzeugmaschinen und dem Werkzeugmaschinenbau (Maschinenkomponenten, Funktionalitäten, Anwendungs- /

		Einsatzmöglichkeiten) sowie zu Montagetechnologien und Verbindungstechniken (Auslegung von Verbindungen, prozesstechnische Umsetzung und Realisierung) vermittelt. Einen weiteren Schwerpunkt stellen der Elektromaschinenbau und die Elektronikproduktion (Funktionsweise und Herstellung von elektronischen Antriebseinheiten, Auslegung und Herstellung von elektronischen Komponenten) dar.
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen • Die Studierenden erwerben grundlegende Kenntnisse in der Metallkunde und der Verarbeitung von Metallen. • Die Studierenden erhalten einen Überblick über die Produktionsverfahren Urformen, Umformen, Fügen, Trennen, ihre Untergruppen • Die Studierenden erwerben ein grundlegendes Prozessverständnis hinsichtlich der wirkenden Mechanismen. • Die Studierenden erwerben Wissen über die Prozessführung sowie spezifische Eigenschaften der Produktionsverfahren. • Die Studierenden erwerben grundlegendes Verständnis zu den Eigenschaften von Kuststoffen und deren Verarbeitung • Die Studierenden erwerben Kenntnisse über werkstoffwissenschaftliche Aspekte und Werkstoffeigenschaften sowie Werkstoffverhalten vor und nach den jeweiligen Bearbeitungsprozessen • Die Studierenden erwerben fundamentale Kenntnisse zu Multi-Materialien-Verbunden. • Die Studierenden erlangen grundlegende Kenntnisse zur Funktionsweise von elektrischen Antriebseinheiten und deren Herstellung sowie die Herstellung von elektrischen Komponenten (MID) • Die Studierenden erhalten grundlegende Kenntnisse im Bereich der Produktentwicklung und Produktauslegung (Verfahrensmöglichkeiten, Verfahrensgrenzen, Designeinschränkungen, etc.) Verstehen • Die Studierenden sind in der Lage die grundlegenden Prinzipien von Fertigungsprozessen und der Systemauslegung zu verstehen • Die Studierenden verstehen die Grundlagen des Anlagen- und Werkzeugbaus Anwenden • ° Die Studierenden können die verschiedenen Fertigungsverfahren erkennen und normgerecht differenzieren
7	**Voraussetzungen für die Teilnahme**	Keine

8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Fertigungsautomatisierung und Produktionstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	2 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Nebenfach Maschinenbau: Technische Mechanik

1	**Modulbezeichnung** 94500	**Dynamik starrer Körper** Dynamics of rigid bodies	**7,5 ECTS**
2	Lehrveranstaltungen	Tutorium: DSK (Tut) (2 SWS) Vorlesung: Dynamik starrer Körper (3 SWS) Übung: Übung zur Dynamik starrer Körper (2 SWS)	- 7,5 ECTS -
3	Lehrende	David Holz Eduard Sebastian Scheiterer Prof. Dr.-Ing. Sigrid Leyendecker Dr.-Ing. Giuseppe Capobianco Xiyu Chen	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Sigrid Leyendecker
5	**Inhalt**	• Kinematik von Punkten und starren Körpern • Relativkinematik von Punkten und starren Körpern • Kinetik des Massenpunktes • Newton'sche Axiome • Energiesatz • Stoßvorgänge • Kinetik des Massenpunktsystems • Lagrange'sche Gleichungen 2. Art • Kinetik des starren Körpers • Trägheitstensor • Kreiselgleichungen • Schwingungen
6	**Lernziele und Kompetenzen**	Die Studierenden • sind vertraut mit den grundlegenden Begriffen und Axiomen der Dynamik; • können Bewegungen von Massepunkten und starren Körpern in verschiedenen Koordinatensystemen beschreiben; • können die Bewegungsgleichungen von Massepunkten und starren Körpern mittles der Newtonschen Axiome oder mittels der Lagrangeschen Gleichungen aufstellen; • können die Bewegungsgleichungen für einfache Stoßprobleme lösen; • können die Bewegungsgleichung für einfache Schwingungsprobleme analysieren.
7	**Voraussetzungen für die Teilnahme**	Kenntnisse aus dem Modul |"Statik, Elastostatik und Festigkeitslehre"| bzw. |"Statik und Festigkeitslehre"|
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Technische Mechanik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)

12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 105 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Gross, Hauger, Schnell, Wall: Technische Mechanik 3, Berlin:Springer, 2006

1	**Modulbezeichnung** 94660	**Statik und Festigkeitslehre** Statics and mechanics of materials	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Statik und Festigkeitslehre (3 SWS) Sonstige Lehrveranstaltung: Tutoreneinführung zur Statik und Festigkeitslehre (2 SWS) Tutorium: Übung Statik und Festigkeitslehre (2 SWS)	- - -
3	Lehrende	Prof. Dr.-Ing. Silvia Budday Lucie Spannraft Dr.-Ing. Gunnar Possart Michael Lengger	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Kai Willner
5	**Inhalt**	• Kraft- und Momentenbegriff, Axiome der Statik • ebene und räumliche Statik • Flächenmomente 1. und 2. Ordnung • Haft- und Gleitreibung • Spannung, Formänderung, Stoffgesetz • überbestimmte Stabwerke, Balkenbiegung • Torsion • Elastizitätstheorie und Festigkeitsnachweis • Stabilität
6	**Lernziele und Kompetenzen**	Wissen Die Studierenden kennen • die axiomatischen Grundlagen der Technischen Mechanik sowie die entsprechenden Fachtermini. • das Schnittprinzip und die Einteilung der Kräfte in eingeprägte und Reaktionskräfte bzw. in äußere und innere Kräfte. • die Gleichgewichtsbedingungen am starren Körper. • das Phänomen der Haft- und Gleitreibung. • die Begriffe der Verzerrung und Spannung sowie das linear-elastische Stoffgesetz. • den Begriff der Hauptspannungen sowie das Konzept der Vergleichsspannung und Festigkeitshypothesen. • das Problem der Stabilität und speziell die vier Eulerschen Knickfälle für ein schlankes Bauteil unter Drucklast. Verstehen Die Studierenden • können Kräfte nach verschiedenen Kriterien klassifizieren. • können verschiedene Lagerungsarten unterscheiden und die entsprechenden Lagerreaktionen angeben. • können den Unterschied zwischen statisch bestimmten und unbestimmten Systemen erklären. • können den Unterschied zwischen Haft- und Gleitreibung erläutern. • können das linear-elastische, isotrope Materialgesetz angeben und die Bedeutung der Konstanten erläutern. • können die Voraussetzungen der Euler-Bernoulli-Theorie schlanker Balken erklären.

		• verstehen die Idee der Vergleichsspannung und können verschiedene Festigkeitshypothesen erklären. Anwenden Die Studierenden können • den Schwerpunkt eines Körpers bestimmen. • ein System aus mehreren Körpern geeignet freischneiden und die entsprechenden eingeprägten Kraftgrößen und die Reaktionsgrößen eintragen. • für ein statisch bestimmtes System die Reaktionsgrößen aus den Gleichgewichtsbedingungen ermitteln. • die Schnittreaktionen für Stäbe und Balken bestimmen. • die Spannungen im Querschnitt schlanker Bauteile (Stab, Balken) unter verschiedenen Belastungen (Zug, Biegung, Torsion) ermitteln. • die Verformungen schlanker Bauteile ermitteln. • aus einem gegebenen, allgemeinen Spannungszustand die Hauptspannungen sowie verschiedene Vergleichsspannungen ermitteln. • die kritische Knicklast für einen gegebenen Knickfall bestimmen. Analysieren Die Studierenden können • ein geeignetes Modell für schlanke Bauteile anhand der Belastungsart und Geometrie auswählen. • ein problemangepasstes Berechnungsverfahren zur Ermittlung von Reaktionsgrößen und Verformungen auch an statisch unbestimmten Systemen wählen. • eine geeignete Festigkeitshypothese wählen. • den relevanten Knickfall für gegebene Randbedingungen identifizieren. Evaluieren (Beurteilen) Die Studierenden können • den Spannungszustand in einem Bauteil hinsichtlich Aspekten der Festigkeit bewerten. • den Spannungszustand in einem schlanken Bauteil hinsichtlich Aspekten der Stabilität bewerten.
7	**Voraussetzungen für die Teilnahme**	Alle Informationen zum Ablauf der Lehrveranstaltung werden über den StudOn-Kurs kommuniziert. Deshalb bitten wir Sie, sich unter https://www.studon.fau.de/cat5282.html einzuschreiben. Der Beitritt ist nicht, wie sonst üblich, passwortgeschützt, sondern erfolgt nach Bestätigung durch den Dozenten. Dies geschieht mitunter nicht umgehend, aber rechtzeitig vor dem ersten Termin. Wir bitten um Ihr Verständnis.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!

9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Technische Mechanik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 105 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Gross, Hauger, Schnell, Wall: Technische Mechanik 1, Berlin:Springer 2006 • Gross, Hauger, Schnell, Wall: Technische Mechanik 2, Berlin:Springer 2007

Nebenfach Maschinenbau: Qualitätsmanagement und Messtechnik

1	**Modulbezeichnung** 94510	**Grundlagen der Messtechnik** Fundamentals of metrology	**5 ECTS**
2	Lehrveranstaltungen	Übung: Grundlagen der Messtechnik - Übung (2 SWS) Vorlesung mit Übung: Grundlagen der Messtechnik (2 SWS)	- 5 ECTS
3	Lehrende	Prof. Dr.-Ing. Tino Hausotte	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Tino Hausotte
5	**Inhalt**	**Inhalt (Vorlesung):** • Allgemeine Grundlagen • Was ist Metrologie: Metrologie und Teilgebiete, Einsatzbereiche, historische Entwicklung des Einheitssystems, SI-Einheitensystem SI-Einheiten (cd, K, kg, m, s, A, mol) Größe, Größenwert Extensive und intensive Größen Messung, Messgröße, Maßeinheit, Messergebnis, Messwert, Gebrauch und korrekte Angabe der Einheiten, Schreibweisen von Größenwerten, Angabe von Einheiten Grundvoraussetzungen für das Messen Rückführung der Einheiten • Messprinzipien, Messmethoden und Messverfahren: Messprinzip, Messmethode, Messverfahren Einteilung der Messmethoden, Ausschlagmessmethode, Differenzmessmethode, Substitutionsmessmethode und Nullabgleichsmethode (Kompensationsmethode) Prinzip eines Messgerätes, direkte und indirekte Messmethoden Kennlinie und Kennlinienarten, analoge und digitale Messmethoden, kontinuierliche und diskontinuierliche Messung, Auflösung, Empfindlichkeit, Messbereich absolute und inkrementelle Messmethoden • Statistik Auswertung von Messreihen: Berechnung eines Messergebnisses anhand von Messreihen Grundbegriffe der deskriptiven Statistik Darstellung und Interpretation von Messwertverteilungen (Histogramme) Häufigkeit (absolute, relative, kumulierte, relative kumulierte) Berechnung und Interpretation grundlegender Parameter: Lage (Mittelwert, Median, Modus), Streuung (Spannweite, Varianz, Standardabweichung, Variationskoeffizient), Form (Schiefe, Kurtosis bzw. Exzess) Grundbegriffe der Stochastik, Wahrscheinlichkeiten, Verteilungen (Rechteck-, U- und Normalverteilung), Zentraler Grenzwertsatz, statistische Momente Grundbegriffe der analytischen Statistik, statistische Tests und statistische Schätzverfahren Korrelation und Regression • Messabweichungen und Messunsicherheit: Messwert, wahrer Wert, Ringvergleich, vereinbarter Wert Einflüsse auf die Messung (Ishikawa-Diagramm) Messabweichung (absolute, relative, systematische, zufällige) Umgang mit Messabweichungen, Korrektion bekannter systematischer Messabweichungen Kalibrierung, Verifizierung, Eichung

		Messpräzision, Messgenauigkeit, Messrichtigkeit Wiederholbedingungen/-präzision, Vergleichsbedingungen/-präzision, Erweiterte Vergleichsbedingungen/-präzision Fehlerfortpflanzungsgesetz (altes Konzept), Messunsicherheit, Eigenunsicherheit, Übersicht über Standardverfahren des GUM (Messunsicherheit), korrekte Angabe eines Messergebnisses • Messgrößen des SI-Einheitensystems • Messen elektrischer Größen und digitale Messtechnik: SI-Basiseinheit Ampere, Widerstands- und Spannungsnormale, Messung von Strom und Spannung, Lorentzkraft, Drehspulmesswerk, Bereichsanpassung Widerstandsmessung, strom- und spannungsrichtige Messung, Wheatstonesche Brückenschaltung (Viertel-, Halb- und Vollbrücke, Differenzmethode und Kompensationsmethode) Charakteristische Werte sinusförmiger Wechselgrößen, Dreheisenmesswerk, Wechselspannungsbrücke Messsignale, dynamische Kennfunktionen und Kennwerte, Übertragungsfunktionen (Frequenzgänge) Digitalisierungskette, Zeit- und Wertdiskretisierung, Alias-Effekte, Shannons Abtasttheorem, Filter, Operationsverstärker (Invertierender Verstärker, Nichtinvertierender Verstärker, Impedanzwandler, invertierender Addierer, Differenzverstärker, Integrierer, Differenzierer, Instrumentenverstärker), Abtast-Halte-Glied, Analog-Digital-Wandlung, Abweichungen bei der Analog-Digital-Wandlung Universelle Messgeräte (Digitalmultimeter, analoge und digitale Oszilloskope) • Messen optischer Größen: Licht und Eigenschaften des Lichtes Empfindlichkeitsspektrum des Auges Radiometrie und Photometrie SI-Basiseinheit Candela (cd, Lichtstärke) Strahlungsfluss, radiometrisches (fotometrisches) Grundgesetz, photometrische und radiometrische Größen Strahlungsgesetze Fotodetektoren (Fotowiderstände, Fotodioden, Betriebsarten, Bauformen, CCD- und CMOS-Sensoren) • Messen von Temperaturen: Temperatur, SI-Basiseinheit Kelvin, Definition, Wärmeübertragung (Wärmeleitung, Konvektion, Wärmestrahlung) Thermodynamische Temperatur Primäre und sekundäre Temperaturmessverfahren, praktische Temperaturskalen, Fixpunkte (Tripelpunkte, Erstarrungspunkte), Fixpunktzellen, klassische Temperaturskalen, internationale Temperaturskala (ITS-90) Berührungsthermometer, thermische Messabweichungen, thermische Ausdehnung, Gasthermometer, Flüssigkeitsglasthermometer, Bimetall-Thermometer, Metall-Widerstandsthermometer (Kennlinie, Genauigkeit, Bauformen, Messschaltungen), Thermoelemente (Seebeck-Effekt, Bauformen, Ausgleichsleitungen,

		Messschaltungen) Strahlungsthermometer (Prinzip, Strahlungsgesetze, Pyrometer, Messabweichungen) • Zeit und Frequenz: SIBasiseinheit Sekunde, Zeitmessung (Aufgaben, Historie, mechanische Uhren, Quarzuhren, Atomuhr) Darstellung der Zeit Verbreitung der Zeitskala UTC Globales Positionssystem (GPS) Frequenz- und Phasenwinkelmessung • Längenmesstechnik: SIBasiseinheit Meter Messschieber, Abbesches Komparatorprinzip, Bügelmessschraube, Abweichungen 1.- und 2.-Ordnung Längenmessung mit Linearencodern (Bewegungsrichtung, Ausgangssignale, Differenzsignale, Demodulation) Absolutkodierung (V-Scannen und Gray Code) Interferometrie, Michelson-Interferometer, transversale elektromagnetische Wellen, Grundlagen der Interferenz, destruktive und konstruktive Interferenz, Homodynprinzip, Heterodynprinzip, Interferenz am Homodyninterferometer, Demodulation am Homodyn- und Heterodyninterferometer, Einfluss Luftbrechzahl, Realisierung der Meterdefinition, Reflektoren und Aufbau von Interferometern, induktive Längenmessung, kapazitive Längenmessung, Laufzeitmessung • Masse, Kraft und Drehmoment: SIBasiseinheit Kilogramm, Definition Masse, Kraft und Drehmoment Massenormale (Vergleiche, Bauformen und Abweichungsgrenzen), Prinzip der Masseableitung, Stabilität der Einheit und Neudefinition Messprinzipien von Waagen, Einflussgrößen bei Massebestimmung (lokale Erdbeschleunigung, Luftauftrieb), Balkenwaage (unterschalige Waagen, Empfindlichkeit, Bauformen, oberschalige Waagen, Ecklastabhängigkeit), Federwaage, DMS, Verformungskörper, DMS-Waage, EMK-Waage, Massekomparatoren Drehmomentmessung (Reaktions- und Aktionsdrehmoment) • Teilgebiete der industriellen Messtechnik • Prozessmesstechnik: Messgrößen der Prozessmesstechnik Definition des Druckes, Druckarten (Absolutdruck, Überdruck, Differenzdruck) Druckwaage (Kolbenmanometer), U-Rohrmanometer und -Barometer, Rohrfedermanometer, Plattenfedermanometer Drucksensoren (mit DMS, piezoresistiv, kapazitiv, piezoelektrisch) Durchflussmessung (Volumenstrom und Massestrom, Strömung von Fluiden) volumetrische Verfahren, Wirkdruckverfahren, magnetisch-induktive Durchflussmessung, Ultraschall-Durchflussmessung Massedurchflussmessung (Coriolis, thermisch) • Fertigungsmesstechnik: Aufgaben, Methoden, Ziele und Bereiche der Fertigungsmesstechnik Gestaltparameter von Werkstücken (Mikro- und Makrogestalt), Geometrische Produktspezifikation (GPS), Gestaltabweichungsarten Geräte und Hilfsmittel der Fertigungsmesstechnik, Gegenüberstellung klassische Fertigungsmesstechnik

und Koordinatenmesstechnik, Auswertung Bauarten und Grundstruktur von Koordinatenmessgeräten Vorgehensweise bei Messen mit einem Koordinatenmessgerät

Inhalt (Übung):

- Grundlagen der Elektrotechnik (Wiederholung von Grundlagen)
- Statistik Auswertung von Messreihen (Histogramme, Hypothesentest, Konfidenzintervalle, statistischen Maßzahlen)
- Korrelation und Regression (Korrelationskoeffizient, Fehlerfortpflanzung, Residuenanalyse)
- Messabweichungen, Einführung in die Messunsicherheitsberechnung (Kompensation systematischer Abweichungen, Messunsicherheitsanalyse einer einfachen Messung)
- Elektrische Größen, Messelektronik und Analog-Digital-Umsetzung (Abweichungsberechnung bei der Strommessung, Anpassungsnetzwerk für ein Drehspulinstrument, Bereichsanpassung mit einem Operationsverstärker)
- Anwendung der Wheatstoneschen Brückenschaltung bei Messungen mit Dehnungsmessstreifen
- Messungen mit Fotodioden bei unterschiedlichen Betriebsarten
- Temperaturmesstechnik (Aufgaben zu Metall-Widerstandsthermometern und Pyrometern)
- Längenmesstechnik (Abbesche Prinzip, Induktivität eines Eisenkerns mit Luftspalt, Foliendickenmessung mittels einer kapazitiven Messeinrichtung)
- Messen von Kraft und Masse (Massewirkung, Balkenwaage, Federwaage, piezoelektrischer Kraftsensor)
- Prozessmesstechnik (Druck- und Durchflussmessung, U-Rohrmanometer, Corioliskraftmessung, Ultraschallmessverfahren, Turbinenzähler)
- Fertigungsmesstechnik (Standardgeometrieelemente, Angabe von Toleranzen, Prüfen von Rundheitsabweichungen mit Hilfe eines Feinzeigers)

Contents:

- General basics
- What is metrology: Metrology and braches, application fields, historical development of the unit system, SI unit system Definitions of SI units (cd, K, kg, m, s, A, mol) Quantity, quantity value Extensive and intensive quantities Measurement, measurand, measurement unit, measurement result, measured quantity value Correct use and notation of units and of quantity values Basic requirements for the measurement Traceability
- Principles, methods and procedures of measurement: Principles, methods and procedures of measurement Classification of measurement methods, deflection, differential, substitution and compensation measurement methods

Principle of a measuring instrument, direct and indirect measurement methods Characteristic curve, types of characteristic curves, analogue and digital measurement methods, continuous and discontinuous measurement, resolution, sensitivity, measuring interval Absolute and incremental measurement methods

- Statistics Evaluation of measurements series: Calculation of a measurement result based on measurement series Basic terms of descriptive statistics Presentation and interpretation of measured value distributions (histograms) Frequency (absolute, relative, cumulative, relative cumulative) Calculation and interpretation of basic parameters: location (mean, median, mode), dispersion (range, variance, standard deviation, coefficient of variation), shape (skewness, excess, kurtosis) Basic terms of stochastics, probabilities, distributions (rectangle, U and normal distribution), central limit theorem, statistical moments Basic terms of analytical statistics, statistical tests and statistical estimation methods Correlation and regression
- Measurement errors and measurement uncertainty: Measured value, true value, key comparison, conventional quantity value Influences on the measurement (Ishikawa diagram) Measurement error (absolute, relative, systematic, random) Handling of errors, correction of known systematic measurement errors Calibration, verification, legal verification Measurement precision, accuracy and trueness Repeatability conditions and repeatability, intermediate precision condition and measurement precision, reproducibility condition of measurement and reproducibility Error propagation law (old concept), measurement uncertainty, definitional uncertainty, overview of standard method of the GUM (measurement uncertainty), correct specification of a measurement result
- Mesurands of the SI system of units
- Measurement of electrical quantities: SI base unit Ampere, resistance and voltage standards, measurement of current and voltage, Lorentz force, moving coil instrument, range adjustment Resistance measurement, current and voltage correct measurement, Wheatstone bridge circuit (quarter, half and full bridge, differential method and compensation method) Characteristic values of sinusoidal alternating quantities, moving iron instrument, alternating voltage bridge Measuring signals, dynamic characteristic functions and characteristics, transfer functions (frequency responses) Digitalisation chain, time and value discretization, aliasing, Shannons sampling theorem, filter, operational amplifier (inverting amplifier, non-inverting amplifier, impedance converter, inverting summing amplifier, differential amplifier, integrating amplifier, differentiating amplifier, instrumentation amplifier), sample-and-hold device, analogue-digital conversion, errors of

analogue-to-digital conversion Universal measuring devices (digital multimeter, analogue and digital oscilloscopes)

- Measurement of optical quantities: Light and properties of light Sensitivity spectra of the eye Radiometry and photometry SI base unit candela (cd, luminous intensity) Radiant flux, radiometric (photometric) fundamental law, photometric and radiometric quantities Radiation laws Photo detectors (photo resistors, photo diodes, modes of operation, designs, CCD and CMOS sensors)
- Measurement of temperatures: Temperature, SI base unit Kelvin, definition, heat transfer (conduction, convection, radiation) Thermodynamic temperature Primary and secondary temperature measurement methods, practical temperature scales, fixpoints (triple points, freezing points), fixpoint cells, classical temperature scales, International Temperature Scale (ITS-90) Contact thermometers, thermal measurement errors, thermal expansion, gas thermometer, liquid thermometer, bimetal thermometer, metal resistance thermometers (characteristic curve, accuracy, designs, circuits), thermocouples (Seebeck effect, designs, extension wires, measurement circuits) Radiation thermometer (principle, radiation laws, pyrometers, measurement errors)
- Time and frequency: SI base unit second, time measurement (tasks, history, mechanical clocks, quartz clock, atomic clock) Representation of time Propagation of UTC Global Positioning System (GPS) Frequency and phase angle measurement
- Length: SI base unit metre Calliper, Abbe comparator principle, micrometer, errors 1st and 2nd order Length measurement with linear encoders (motion direction, output signals, differential signals, demodulation) Absolute coding (V-Scan and Gray code) Interferometry, Michelson interferometer, transversal electromagnetic waves, basics of interference, destructive and constructive interference, homodyne principle, heterodyne principle, interference on homodyne interferometer, demodulation at homodyne and heterodyne interferometer, influence of air refractive index, realisation of the metre definition, reflectors and assembly of interferometers, inductive length measurement, capacitive length measurement, time of flight measurement
- Mass, force and torque: SIbase unit kilogram, definition of mass, force and torque Mass standards (comparisons, types, deviation limits), principle of mass dissemination, stability of the unit and redefinition Measurement principles of weighing, influences for mass determination (local gravitational acceleration, air buoyancy), beam balance (hanging pan balances, sensitivity, types, top pan balances, corner load sensitivity), spring balance, DMS, deformation elements, DMS balance, EMC balance, mass comparators Measurement of torque (reactive and active)

		• Branches of industrial metrology • Process measurement technology: Quantities of process measurement technology Definition of pressure, pressure types (absolute pressure, overpressure, differential pressure) Deadweight tester (piston manometer), U-tube manometer and barometer, bourdon tube gauge, diaphragm pressure gauge Pressure sensors (with DMS, piezoresistive, capacitive, piezoelectric) Flow measurement (volume flow and mass flow, flow of fluids) Volumetric method, differential pressure method, magneto-inductive flowmeter, ultrasonic flow measurement Mass flow rate measurement (Coriolis, thermal) • Manufacturing metrology: Tasks, methods, objectives and branches of manufacturing metrology Form parameters of workpieces (micro-and macro-shape), geometrical product specification (GPS), geometrical tolerances Comparison of classical manufacturing metrology and coordinate metrology, evaluation Designs and basic structure of coordinate measuring machines Procedure for measuring with a coordinate measuring machine

6	**Lernziele und Kompetenzen**	*Wissen*
		• Die Studierenden kennen grundlegende statistische Methoden zur Beurteilung von Messergebnissen und Ermittlung von Messunsicherheiten.
		• Die Studierenden kennen grundlegende Messverfahren zur Erfassung der Messgrößen aller SI-Einheiten.
		• Die Studierenden kennen das Basiswissen zu Grundlagen der Messtechnik und messtechnischen Tätigkeiten.
		• Die Studierenden haben Grundkenntnisse zur methodisch-operativen Herangehensweise an Aufgaben des Messens statischer Größen, zum Lösen einfacher Messaufgaben und zum Ermitteln von Messergebnissen aus Messwerten.
		Verstehen
		• Die Studierenden können die Eigenschaften von Messeinrichtungen und Messprozessen beschreiben.
		• Die Studierenden können das Internationale Einheitensystem und die Rückführung von Messergebnissen beschreiben.
		Anwenden
		• Die Studierenden können einfache Messungen statischer Größen durchführen.
		• Die Studierenden können Messunsicherheiten komplexer Messeinrichtungen bei gegebenen Eingangsgrößen berechnen.
		Evaluieren (Beurteilen)
		•
		  ° The students know basic statistical methods for the evaluation of measurement results and the determination of measurement uncertainties.
		  ° The students know basic measuring methods for the record of measured values for all SI units.

		° The students have basic knowledge of fundamentals of metrology and metrology activities. ° The students have fundamental knowledge for methodological and operational approach to measuring tasks of static measurement types, to solve basic measurement tasks and to establishing measurement results from measurement values. ° The students are able to describe the characteristics of measuring instruments and measurement processes. ° The students are able to describe the international system of units (SI) and the traceability of measurement results ° The students are able to run basic measurements of static measurands. *Evaluating* The students are able to evaluate measuring systems, measurement processes and measurement results. Students are able to calculate the measurement uncertainty of complex measuring systems for given input variables.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Qualitätsmanagement und Messtechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	International Vocabulary of Metrology Basic and General Concepts and Associated Terms, VIM, 3rd edition, JCGM 200:2008, http://www.bipm.org/en/publications/guides/vim.html DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Korrigierte Fassung 2012, Beuth Verlag GmbH, 4. Auflage 2012 Hoffmann, Jörg: Handbuch der Messtechnik. 4. Auflage, Carl Hanser Verlag München, 2012 ISBN 978-3-446-42736-5

		Lerch, Reinhard: Elektrische Messtechnik. 6. Auflage, Springer-Verlag Berlin Heidelberg, 2012 ISBN 978-3-642-22608-3 Richter, Werner: Elektrische Meßtechnik. 3. Auflage, Verlag Technik Berlin, 1994 - ISBN 3-341-01106-4 Kohlrausch, Friedrich: Praktische Physik : zum Gebrauch für Unterricht, Forschung und Technik. Band 1-3, 24. Auflage, Teubner Verlag, 1996 ISBN 3-519-23001-1, 3-519-23002-X, 3-519-23000-3 H. Czichos (Hrsg.): Das Ingenieurwissen Gebundene. 7. Auflage, Springer Verlag, 2012, ISBN 978-3-642-22849-0. Ernst, Alfons: Digitale Längen- und Winkelmesstechnik. 4. Auflage, Verlag Moderne Industrie, 2001 ISBN 3-478-93264-5 Pfeifer, Tilo: Fertigungsmeßtechnik. R. Oldenbourg Verlag München Wien, 1998 ISBN 3-486-24219-9 Keferstein, Claus P.: Fertigungsmesstechnik. 7. Auflage, Vieweg +Teubner Verlag, 2011 ISBN 978-3-8348-0692-5 Warnecke, H.-J.; Dutschke, W.: Fertigungsmeßtechnik. Springer-Verlag Berlin Heidelberg New York Tokyo, 1984 ISBN 3-540-11784-9

Schwerpunkt Physiologie

1	**Modulbezeichnung** 22800	**Anatomie und Physiologie für Nichtmediziner** Anatomy and physiology for non-medical students	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Grundlagen der Anatomie und Physiologie für Medizintechniker, Naturwissenschaftler und Ingenieure (2 SWS)	-
3	Lehrende	Dr. Jana Dahlmanns Prof. Dr. Peter Soba Prof. Dr. Christian Alzheimer	

4	**Modulverantwortliche/r**	Dr. Jana Dahlmanns apl.Prof.Dr. Clemens Forster
5	**Inhalt**	• Wissensvermittlung zu Grundlagen der Anatomie, Physiologie und Pathophysiologie • Wissensvermittlung von wichtigen medizinischen Fachbegriffen • Wissensvermittlung von relevanten und häufigen Krankheitsbildern • Wissensvermittlung von relevanten Methoden beim biologischen und technischen Sehen • Diskussion von Methoden und Theorieansätzen, um relevante medizinische Fragestellungen erkennen zu können • Kritische Betrachtung von den wichtigsten bildgebenden Verfahren in wichtigen Krankheitsbildern • Darstellung der Organisationsstrukturen von diagnostischen Prozessen
6	**Lernziele und Kompetenzen**	Die Studierenden • verstehen die wichtigsten und häufigsten medizinische Fachbegriffe • sind vertraut mit den Grundlagen der Anatomie und der Physiologie • kennen wichtige Krankheitsbilder • verstehen und erklären medizinische Fragestellungen in der Diagnostik und Therapie anhand von Beispielen
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Medizinische Informatik Bachelor of Science Informatik 20222 Schwerpunkt Physiologie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	schriftlich (60 Minuten)
11	**Berechnung der Modulnote**	schriftlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	2 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Nebenfach Maschinenbau: Computerintegrierte Produktion

1	**Modulbezeichnung** 95340	**Automotive Engineering I** Automotive engineering	**2,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Automotive Engineering 1 (2 SWS)	2,5 ECTS
3	Lehrende	Prof. Dr.-Ing. Jörg Franke Jan Fröhlich Jean-Marc Gales Dr.-Ing. Alexander Kühl	

4	**Modulverantwortliche/r**	
5	**Inhalt**	Die Vorlesung ist an alle ingenieurwissenschaftliche Studiengänge und Studierenden mit Interesse an einer Tätigkeit in der Automobilindustrie oder deren Umfeld gerichtet. Es werden die Themen der Produktentstehung bis zur Fertigung und Vertrieb beleuchtet. Dabei wird der Aspekt des interdisziplinären Agierens aus unterschiedlichen Blickwinkeln dargestellt. Zum einen werden Einblicke in die technische, konstruktive Umsetzung von wesentlichen Elementen eines Automobils gestreift, zum anderen sollen aber auch strategische und betriebswirtschaftlich bestimmende Größen vermittelt und deren Bedeutung für den Ingenieur vertieft werden. Ziel ist es ein Gesamtverständnis für den Komplex der Automobilindustrie zu vermitteln. Das Automobil ist zunehmend eines der komplexesten Industriegüter. Es ist geprägt durch gesellschaftliche Anforderungen, gesetzliche Restriktionen und unterschiedlichste Markt- und Kundenwünschen weltweit. Lernen Sie die Herausforderungen für die Ingenieurwissenschaften in der Automobilindustrie kennen, die Zusammenhänge verstehen und die Lösungen zu erarbeiten. Folgende thematischen Schwerpunkte werden in der Vorlesung behandelt: • Überblick über die Abläufe und Rahmenbedingungen für die Entwicklung in der Automobilindustrie. • Die Produktentstehung • Der Produktionsprozess in der Automobilindustrie • Integrierte Absicherung • Handelsorganisation: Markteinführung, Marketingkonzepte, Service und Aftermarket Strategien • Elektrifizierung, Hybrid, alternative Antriebe • Elektronik im Fahrzeug: Fahrerassistenz, Navigation, Kommunikation • Neue Technologien für die Herstellung von Karosserien • Passive und aktive Sicherheit. Trend und Markttendenzen, technische Lösungen • Entwicklung der Fahrdynamik • IT-Systeme in der Automobilindustrie • Spitzenleistungen als faszinierende Herausforderungen (Designstudien, Experimentalfahrzeuge, Rennsport) • Qualitätsmanagement

6	**Lernziele und Kompetenzen**	Nach besuch der Vorlesung sind die Studierenden in der Lage: • Einen Überblick über die Produktentstehung bin hin zur Serienentwicklung zu geben • Die Produktionsprozesse im Automobilbau zu verstehen • Supportprozesse wie die integrierte Absicherung zu verstehen • Die Vor- und Nachteile der unterschiedlichen Antriebstechnologien zu nennen • Einen Überblick von Elektrik und Elektronik im Fahrzeug zu haben • Einflüsse auf die Fahrzeugdynamik zu verstehen
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 45 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 95270	**Die Werkzeugmaschine als mechatronisches System** Machine tools as a mechatronic system	**2,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Die Werkzeugmaschine als mechatronisches System (2 SWS)	2,5 ECTS
3	Lehrende	Sebastian Anders Eva Russwurm Prof. Dr.-Ing. Siegfried Russwurm	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Siegfried Russwurm
5	**Inhalt**	• Bedeutung der Mechatronik im Werkzeugmaschinenbau • Grundlegende Begrifflichkeiten mit Bezug auf den Werkzeugmaschinenbau zu den Themen Mechanik, Elektrotechnik und Software • Analyse, Modellierung und Regelung von Werkzeugmaschinen • CNC-Steuerungstechnik für die Werkzeugmaschine • Parallelkinematik-Maschinen • Evolution der Drehmaschinen • Vertikale und horizontale IT-Integration
6	**Lernziele und Kompetenzen**	Nach Absolvierung des Moduls sind die Studierenden in der Lage: • wesentliche mechatronische Komponenten der Werkzeugmaschine zu benennen und zu erläutern. • Modellversuche zur elektrischen Antriebstechnik durchzuführen. • eine analytische Vorgehensweise zur regelungstechnischen Modellbildung anzuwenden. • Regelungstechnische Möglichkeiten der elektrischen Antriebstechnik darzustellen. • die CNC Verfahrenskette vom CAD-Geometriemodell zur Werkzeugposition zu erklären. • Konsequenzen alternativer Maschinenkonzepte (Parallelkinematiken, modulare Maschinen) zu erläutern. • Werkzeugmaschinen als IT-Komponenten (horizontale und vertikale Integration und Kommunikation) darzustellen. • Mechatronische Systeme im allg. Maschinenbau anzuwenden und die Konzepte der Werkzeugmaschine auf andere Maschinenbau-Applikationen zu übertragen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)

11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 45 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 94951	**Grundlagen der Robotik** Fundamentals of robotics	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Jörg Franke
5	**Inhalt**	Die Veranstaltung Grundlagen der Robotik richtet sich insbesondere an die Studierenden der Informatik, des Maschinenbaus, der Mechatronik, der Medizintechnik sowie des Wirtschaftsingenieurwesens. Im Rahmen der Veranstaltung werden zunächst die Grundlagen der modernen Robotik erläutert und anschließend fachspezifische Grundlagen zur Konzeption, Implementierung und Realisierung von Robotersystemen vermittelt. Hierbei liegt der Fokus neben klassischen Industrierobotern auch auf neuen Robotertechnologien für den Service-, Pflege- und Medizinbereich. Im Rahmen der letzten Vorlesungseinheiten sowie der Übungseinheiten werden dem Hörer weiterhin die Grundlagen des Robot Operating System (ROS) vermittelt und es wird durch praktische Übungen die Arbeit und Roboterprogrammierung mit ROS erlernt. Die Veranstaltung umfasst hierfür die nachfolgenden Themenschwerpunkte: • Bauformen, Begriffe, Definitionen, Historie, rechtliche Grundlagen und Roboterethik • Roboteranwendungen in Industrie, Service, Pflege und Medizin • Sensorik und Aktorik für Robotersysteme • Kinematik und Dynamik verschiedener Roboterbauformen • Steuerung, Regelung und Bahnplanung • Varianten der Roboterprogrammierung • Planung und Simulation von Robotersystemen • Robot Operating System (ROS) • Computer Vision (OpenCV)
6	**Lernziele und Kompetenzen**	Ziel der Vorlesung ist, den Studierenden einen fundierten Überblick über aktuelle Roboterapplikationen zu vermitteln sowie die grundlegenden Bauformen, Begrifflichkeiten und gesetzlichen Rahmenbedingungen vorzustellen. Darauf aufbauen werden die notwendigen technischen Grundlagen moderner Robotersysteme sowie die Programmierung eines Roboters mit ROS erlernt. Nach dem Besuch der Veranstaltung sind die Studierenden in der Lage: • Roboter hinsichtlich ihrer Eigenschaften zu klassifizieren, das für eine vorgegebene Anwendung optimale Robotersystem auszuwählen und hierbei ethische und arbeitsschutzrechtliche Aspekte zu berücksichtigen. • Robotersysteme auszulegen, zu entwickeln und die erforderlichen Bewegungsabläufe zu planen, • die für verschiedene Roboterapplikationen notwendige Sensorik und Aktorik auszuwählen,

		• Robotersysteme durch den Einsatz von Planungs- und Simulationswerkzeugen zu validieren • sowie Roboter mit Hilfe des Robot Operating Systems zu programmieren und zu steuern.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222 Nebenfach Maschinenbau: Fertigungsautomatisierung und Produktionstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 97121	**Handhabungs- und Montagetechnik** Industrial handling and assembly technology	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Jörg Franke
5	**Inhalt**	Im Vertiefungsfach Handhabungs- und Montagetechnik wird die gesamte Verfahrenskette von der Montageplanung bis zur Inbetriebnahme der Montageanlagen für mechanische sowie elektrotechnische Produkte aufgezeigt. Einleitend erfolgt die Darstellung von Planungsverfahren sowie rechnergestützte Hilfsmittel in der Montageplanung. Daran schließt sich die Besprechung von Einrichtungen zur Werkstück- und Betriebsmittelhandhabung in flexiblen Fertigungssytemen und für den zellenübergreifenden Materialfluß an. Desweiteren werden Systeme in der mechanischen Montage von Klein- und Großgeräten, der elektromechanischen Montage und die gesamte Verfahrenskette in der elektrotechnischen Montage diskutiert (Anforderung, Modellierung, Simulation, Montagestrukturen, Wirtschaftlichkeit etc.). Abrundend werden Möglichkeiten zur rechnergestützten Diagnose/Qualitätssicherung und Fragestellungen zu Personalmanagement in der Montage und zum Produktrecycling/-demontage behandelt.
6	**Lernziele und Kompetenzen**	Die Studierenden sind in der Lage: • die Montagefreundlichkeit von Produkten zu beurteilen und zu verbessern, • Montage- und Handhabungsprozesse zu beurteilen, auszuwählen und zu optimieren, • die dazu erforderlichen Geräte, Vorrichtungen und Werkzeuge zu bewerten, und • Montageprozesse sowie -systeme zu konzipieren, zu planen und weiterzuentwickeln. Dieses Wissen ist vor allem in den Bereichen Produktentwicklung, Konstruktion, Produktionsmanagement, Fertigungsplanung, Einkauf, Vertrieb und Management sowie in allen industriellen Branchen (z. B. Automobilbau, Elektrotechnik, Medizintechnik, Maschinen- und Anlagenbau) erforderlich.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222 Nebenfach Maschinenbau: Fertigungsautomatisierung und Produktionstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten)

11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Feldmann, Klaus; Schöppner, Volker; Spur, Günter (Hg.) (2014): Handbuch Fügen, Handhaben, Montieren. 2., vollständig neu bearbeitete Auflage. München: Hanser. • Lotter, Bruno; Wiendahl, Hans-Peter (2012): Montage in der industriellen Produktion. Berlin, Heidelberg: Springer Berlin Heidelberg. • Rainer Müller, Jörg Franke, Dominik Henrich, Bernd Kuhlenkötter, Annika Raatz, Alexander Verl (Hg.) (2019): Handbuch Mensch-Roboter-Kollaboration: Hanser Fachbuchverlag.

1	**Modulbezeichnung** 94947	**Industrie 4.0 - Anwendungsszenarien in Design und Engineering** Industry 4.0 - Application scenarios in design and engineering	**2,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Ulrich Löwen
5	**Inhalt**	Der Industrie-Anlagenbau ist durch hohe technische Komplexität und ein hohes Maß geschäftlicher Risiken gekennzeichnet. Dieses Geschäft hat allerdings für Hochlohnländer wie Deutschland eine strategische Bedeutung: Einerseits ermöglicht die Beherrschung dieser Art von Geschäft die Generierung von nachhaltigen Wettbewerbsvorteilen, da aufgrund der Komplexität ein Kopieren" für Mitbewerber nicht zielführend ist. Andererseits generiert diese Geschäftsart aufgrund der engen Zusammenarbeit mit konkreten Kunden permanent Innovationsideen, welche direkt am Markt eingesetzt und erprobt werden können, sodass dadurch eine Zukunftsorientierung und -sicherung gegeben ist. Allerdings gibt es derzeit keine wissenschaftliche Community, die sich dieser Fragestellung umfassend annimmt. Es ist daher wichtig, den nachwachsenden Generationen von Jungingenieuren die strategische Bedeutung des Themas und mögliche Lösungskonzepte frühzeitig zu vermitteln.
6	**Lernziele und Kompetenzen**	Die Studierenden sollen ein Bewusstsein im Hinblick auf die Potentiale und Risiken des Projektgeschäfts, des Engineerings bzw. der Systemintegration im Kontext von Industrieanlagen entwickeln. Dazu werden branchen- und domänenübergreifende Engineering-Konzepte, -Methoden und -Prozesse vermittelt. Die Vorlesung ist auf Basis der folgenden Leitlinien aufgebaut: • Startpunkt aller Betrachtungen sind jeweils die Treiber aus geschäftlicher und technischer Sicht, die in ihren prinzipiellen Wechselwirkungen untereinander betrachtet werden. Auf dieser Basis werden die Anforderungen an Lösungsansätze bezüglich Geschäftsmodellen, Strategien, Konzepten und Methoden abgeleitet und diskutiert. • Die behandelten Themen werden durch praktische Beispiele aus dem Umfeld des Siemens Konzerns illustriert. Ziel ist dabei, Beispiele aus möglichst unterschiedlichen Geschäften (z.B. Walzwerke, Kraftwerke, Energieübertragung und -verteilung, Logistik, etc.) zu nutzen, um die Gemeinsamkeiten, aber auch Unterschiede transparent zu machen. • Die vorgestellten branchen- und domänenübergreifenden Lösungsansätze in Form von Strategien, Konzepten, Methoden, etc. werden in ein gesamtheitliches Rahmenwerk eingeordnet, um so die Querbezüge und Abhängigkeiten zu verdeutlichen.

		Die Studierenden sind nach Besuch der Lehrveranstaltung in der Lage • die geschäftlichen und technischen Treiber und Herausforderungen im Kontext des Industrieanlagen-Geschäfts umfassend zu verstehen, • grundsätzliche Ansätze der Modellbildung bezüglich Systemen und Prozessen zu unterscheiden und zu nutzen • sowie branchen- und domänenübergreifende Engineering-Konzepte, - Methoden und -Prozesse als Basis für eine konkrete Anwendung beurteilen zu können Das im Rahmen dieser Lehrveranstaltung vermittelte Wissen ist in allen Bereichen der projektbasierten industriellen Branchen, so z. B. im allgemeinen Maschinen-, insbesondere aber im (Groß-) Anlagenbau erforderlich.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) Klausur, Dauer (in Minuten): 60
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 45 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 94946	**Industrie 4.0 - Anwendungsszenarien in Produktion und Service** Industry 4.0 - Application scenarios in production and service	**2,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Industrie 4.0 - Anwendungsszenarien in Produktion und Service (2 SWS)	2,5 ECTS
3	Lehrende	Jonathan Fuchs Prof. Dr. Ulrich Löwen	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Jörg Franke
5	**Inhalt**	Die IT-Durchdringung in der produzierenden Industrie nimmt rasant zu. Der nutzenstiftende Einsatz von IT bei der Gestaltung von Wertschöpfungsprozessen hat für Deutschland eine zentrale strategische Bedeutung. Diese Trends werden unter Begriffen wie Industrie 4.0" und Industrial Internet" bzw. Internet of Things" weltweit diskutiert. Dabei treffen doch recht unterschiedliche Sichtweisen aufeinander. In der Vorlesung werden diese Trends und Visionen anhand von ausgewählten Anwendungsszenarien erläutert. Außerdem werden die dafür zum Verständnis notwendigen Grundlagen erklärt. Ziele: • Bewusstseinsschärfung bezüglich der Auswirkungen der Digitalisierung auf die produzierende Industrie • Verständnis von Geschäftstreibern, technischen Möglichkeiten und deren Wechselwirkungen in der produzierenden Industrie • Vermittlung Branchen- und Domänen-übergreifender Prozesse und Methoden in der produzierenden Industrie
6	**Lernziele und Kompetenzen**	Den Studierenden sollen die Auswirkungen der Digitalisierung auf die produzierende Industrie verdeutlicht und dadurch ein Bewusstsein für diese Entwicklungen geschaffen werden. Zusätzlich soll ein Verständnis für Geschäftstreiber, technische Möglichkeiten und deren Wechselwirkungen in der produzierenden Industrie sowie branchen- und domänenübergreifender Prozesse und Methoden vermittelt werden. Die Vorlesung ist auf Basis der folgenden Leitlinien aufgebaut: • Methodische und konsequente Trennung der Diskussion von Problemperspektive, konzeptioneller Lösungsperspektive und technischer Umsetzungsperspektive • Umfassendes Gesamtverständnis bezüglich der oft sehr vielschichtigen wirtschaftlichen und technischen Zusammenhänge (zu Lasten eines tiefen technischen Detaildiskussion) • Betonung des für einen Anwender gestifteten (geschäftlichen) Nutzens und der möglichen Alleinstellungsmerkmale für einen Standort Deutschland Die Studierenden sind nach Besuch der Lehrveranstaltung in der Lage: • die kontroversen und vielschichtigen Diskussionen im Umfeld der Digitalisierung in der Produzierenden Industrie in einen konsistenten Gesamtkontext einzuordnen

		• anhand repräsentativer Beispiele den Unterschied zu verstehen zwischen dem aktuellen Stand der Technik und Forschung sowie den durch Industrie 4.0 postulierten Innovationshypothesen • aufgrund der vermittelten Beispiele und Methoden durch eine Hinterfragung von Zielen und des wirtschaftlichen Nutzens die oft stark emotional geführten Diskussionen im Kontext von Industrie 4.0 zu versachlichen Das im Rahmen dieser Lehrveranstaltung vermittelte Wissen ist in allen Bereichen der industriellen Branchen, so z. B. im Automobilbau, der Informatik und Wirtschaftsinformatik, der Elektrotechnik und Medizintechnik und dem Maschinen- und Anlagenbau erforderlich.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 35 h Eigenstudium: 40 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 319238	**Industrie 4.0 für Ingenieure** Industry 4.0 for engineers	**2,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Jörg Franke
5	**Inhalt**	Der Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik bietet im Sommersemester die Vorlesung Industrie 4.0 für Ingenieure" als technisches Wahlmodul an. Diese Ringvorlesung wird von renommierten Mitgliedern der Wissenschaftlichen Gesellschaft für Montage, Handhabung und Industrierobotik (MHI, www.wgmhi.de) gehalten, die ausgehend von ihren jeweiligen Fachgebieten in den Themenkomplex Industrie 4.0" einführen. Folgende Themengebiete rund um die Digitalisierung werden unter anderem behandelt: • Industrierobotik • Netzwerk- und Cloudtechnologien • Software und Steuerung • Der Mensch in I4.0 • Industrial Data Science.
6	**Lernziele und Kompetenzen**	Den Studierenden sollen die Auswirkungen und technischen Ausprägungen des Zukunftsprojekts Industrie 4.0 verdeutlicht und dadurch ein Bewusstsein für diese Entwicklungen geschaffen werden. Zusätzlich soll ein Verständnis für Geschäftstreiber, technische Möglichkeiten und deren Wechselwirkungen sowie branchen- und domänenübergreifende Prozesse und Methoden vermittelt werden. Die Studierenden sind nach Besuch der Lehrveranstaltung in der Lage: • die kontroversen und vielschichtigen Diskussionen im Umfeld von Industrie 4.0 in einen konsistenten Gesamtkontext einzuordnen • anhand repräsentativer Beispiele den Unterschied zwischen dem aktuellen Stand der Technik und Forschung sowie den durch Industrie 4.0 postulierten Innovationshypothesen zu verstehen
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222 Nebenfach Maschinenbau: Fertigungsautomatisierung und Produktionstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur Klausur, Dauer (in Minuten): 60

11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 45 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 97123	**Integrated Production Systems** Integrated production systems	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Integrated Production Systems (vhb) (4 SWS)	5 ECTS
3	Lehrende	Prof. Dr.-Ing. Jörg Franke Bernd Hofmann	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Jörg Franke
5	**Inhalt**	• Concepts and Success Factors of Holistic Production Systems • Production organization in the course of time • The Lean Production Principle (Toyota Production System) • The 7 Types of Waste (Muda) in Lean Production • Visual management as a control and management instrument • Demand smoothing as the basis for stable processes • Process synchronization as the basis for capacity utilization • Kanban for autonomous material control according to the pull principle • Empowerment and group work • Lean Automation - "Autonomation" • Fail-safe operation through Poka Yoke • Total Productive Maintenance • Value stream analysis and value stream design • Workplace optimization (lean manufacturing cells, U-Shape, Cardboard Engineering) • OEE analyses to increase the degree of utilization • Quick Setup (SMED) • Implementation and management of the continuous improvement process (CIP, Kaizen) • Overview of quality management systems (e.g. Six Sigma, TQM, EFQM, ISO9000/TS16949) and analysis tools for process analysis and improvement (DMAIC, Taguchi, Ishikawa) • administrative waste • Specific design of the TPS (e.g. for flexible small-batch production) and adapted implementation of selected international corporations
6	**Lernziele und Kompetenzen**	After successfully attending the course, students should be able to • Understand the importance of holistic production systems; • Understand and evaluate Lean Principles in their context; • to evaluate, select and optimise the necessary methods and tools; • To be able to carry out simple projects for the optimisation of production and logistics on the basis of what has been learned in a team.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 2

9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 94920	**International Supply Chain Management** International supply chain management	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: International Supply Chain Management (vhb) (4 SWS)	5 ECTS
3	Lehrende	Daniel Utsch Prof. Dr.-Ing. Jörg Franke	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Jörg Franke
5	**Inhalt**	Contents: The virtual course intents to give an overview on the main tasks of a supply chain manager in an international working environment: • Goals and tasks • Methods and tools • International environment • Knowledge and experience of industrial practice • Cutting edge research on SCM For practical training, 3 additional Case Studies are executed as part of the course. Lehreinheiten / Units: • Integrated logistics, procurement, materials management and production • Material inventory and material requirements in the enterprise • Strategic procurement • Management of procurement and purchasing • In-plant material flow and production systems • Distribution logistics, global tracking and tracing • Modes of transport in international logistics • Disposal logistics • Logistics controlling • Network design in supply chains • Global logistic structures and supply chains • IT systems in supply chain management • Sustainable supply chain management
6	**Lernziele und Kompetenzen**	After having completed this course successfully, the student will be able to • define the basic terms of supply chain management • understand important procurement methods and strategies • name and classify different stock types and strategies • analyse possibilities for cost reduction in supply chains • know and differentiate central IT systems of supply chain management • explain disposal and controlling strategies • recognise the main issues in international supply networks • know the possibilities of transformation to a sustainable supply chain • assess different modes of transport
7	**Voraussetzungen für die Teilnahme**	Keine

8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 95068	**Machine Learning for Engineers II: Advanced Methods** Machine learning for engineers II: Advanced methods	**2,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Machine Learning for Engineers II: Advanced Methods (VHB) (2 SWS)	2,5 ECTS
3	Lehrende	Prof. Dr. Björn Eskofier	

4	**Modulverantwortliche/r**	Thomas Altstidl Prof. Dr. Björn Eskofier
5	**Inhalt**	This is an advanced course with a focus on deep learning (DL) techniques that are typically applied to data science problems. Knowledge is deepened by two practical exercises to gain hands-on experience. The course covers • Extended introduction into fundamental concepts of deep neural networks (DNN) • In-depth review of various optimization techniques for learning neural network parameters • Specification of several regularization techniques for neural networks • Theoretical understanding of application-specific neural network architectures (such as convolutional neural networks (CNN) for images and recurrent neural networks (RNN) for time series)
6	**Lernziele und Kompetenzen**	After successfully participating in this course, students should be able to • discuss advantages and disadvantages of different optimization techniques • design a suitable and promising neural network architecture and train it on existing data using Python and Keras • choose a suitable regularization technique in case of problems
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten) Electronic exam (online), 60min
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 75 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch

16	**Literaturhinweise**	1) Machine Learning: A Probabilistic Perspective, Kevin Murphy, MIT Press, 2012 2) The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani, and Jerome Friedman, Springer, 2009 3) Deep Learning, Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press, 2016

1	**Modulbezeichnung** 95067	**Machine Learning for Engineers I - Introduction to Methods and Tools** Machine learning for engineers I - Introduction to methods and tools	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Machine Learning for Engineers I: Introduction to Methods and Tools (VHB) (0 SWS)	5 ECTS
3	Lehrende	Prof. Dr. Björn Eskofier Prof. Dr.-Ing. Jörg Franke Prof. Dr. Nico Hanenkamp	

4	**Modulverantwortliche/r**	Thomas Altstidl Prof. Dr. Björn Eskofier
5	**Inhalt**	This is an introductary course presenting fundamental algorithms of machine learning (ML) that are typically applied to data science problems. Knowledge is deepened by two practical exercises to gain hands-on experience. The course covers • Introduction to Python programming in the field of data science • Review of typical task domains (such as regression, classification and dimensionality reduction) • Theoretical understanding of widely used machine learning methods (such as linear and logistic regression, support vector machines (SVM), principal component analysis (PCA) and deep neural networks (DNN)) • Practical application of these machine learning methods on engineering problems
6	**Lernziele und Kompetenzen**	After successfully participating in this course, students should be able to • independently recognize the task domain at hand for new applications • select a suitable and promising machine learning methodology based on their known theoretical properties • apply the chosen methodology to the given problem using Python
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur Electronic exam (online), 90min
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 150 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	1) Machine Learning: A Probabilistic Perspective, Kevin Murphy, MIT Press, 2012 2) The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani, and Jerome Friedman, Springer, 2009 3) Deep Learning, Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press, 2016

1	**Modulbezeichnung** 330467	**Multimedia Security** Multimedia security	**5 ECTS**
2	Lehrveranstaltungen	Übung: Multimedia Security Exercises (2 SWS) Vorlesung: Multimedia Security (2 SWS)	- -
3	Lehrende	PD Dr.Ing. Christian Riess	

4	**Modulverantwortliche/r**	PD Dr.Ing. Christian Riess
5	**Inhalt**	This lecture covers a variety of security-related topics around multimedia data. In particular, the lecture presents algorithms and key results from the past 15 years in multimedia security, including topics on image forensics, steganography, watermarking, and biometrics. Selected algorithms are implemented and tested by the participants. It is helpful to bring some knowledge in signal processing or pattern recognition. It is also helpful to be not afraid from equations. Tentative list of topics and algorithms: • Image forensics for manipulation detection in digital media. Statistical and physics-based features for manipulation detection. Detecting traces of manipulations versus validating image authenticity. • Blind source attribution: was an image or video captured with a particular camera? • Steganography for covert communication. Fundamental algorithms, when can their application be detected? • Watermarking for copyright protection in images/video. Fundamental algorithms, and their security. • Biometric features for person re-identification, and practical concerns on their implementation.
6	**Lernziele und Kompetenzen**	Fachkompetenz Verstehen Die Studierenden fassen die wesentlichen Fragestellungen auf dem Gebiet der Multimediasicherheit zusammen (The participants summarize the relevant questions within the field of multimedia security). Die Studierenden nennen und erklären die wesentlichen Fachbegriffe aus den Teilgebieten der Multimediasicherheit (The participants name and explain relevant terms from the subfields of multimedia security). Evaluieren (Beurteilen) Die Studierenden bewerten die Eignung der vorgestellten bildforensischen Algorithmen für ein gegebenes Untersuchungsszenario (The participants evaluate the suitability of the presented image forensics algorithms for a given examination scenario). Erschaffen Die Studierenden implementieren kurze Beispielsprogramme für ausgewählte Algorithmen der Multimediasicherheit (The participants implement short example programs for selected algorithms of multimedia security). Lern- bzw. Methodenkompetenz

		Die Studierenden implementieren ausgewählte Methoden in der Programmiersprache C++ (The participants implement selected methods in the C++ programming language). Sozialkompetenz Die Studierenden implementieren und diskutieren Beispielsmethoden in Gruppenarbeit (The participants implement and discuss the example method in groups). Die Studierenden diskutieren die gesellschaftlichen Auswirkung von Multimediasicherheit am Beispiel aktueller Probleme (The participants discuss multimedia security's impact on society using current issues).
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	begleitend zu der Veranstaltung: • Husrev Sencar, Nasir Memon (Editors): "Digital Image Forensics. There is More to a Picture than Meets the Eye", Springer 2013. • Hany Farid: "Photo Forensics", MIT Press, 2016.

1	**Modulbezeichnung** 97101	**Produktionssystematik** Production systems	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übung zu Produktionssystematik (2 SWS) Vorlesung: Produktionssystematik (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr.-Ing. Jörg Franke Andreas Morello Jonathan Fuchs Martin Barth Sebastian Anders Baris Albayrak David Kunz	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Jörg Franke
5	**Inhalt**	Das Modul Produktionssystematik thematisiert die gesamte Bandbreite der technischen Betriebsführung von der Planung, Organisation und technischen Auftragsabwicklung bis hin zu Fragen des Management und der Personalführung, Entlohnung sowie Kosten- und Wirtschaftlichkeitsrechnung. Im Rahmen dieses Moduls findet eine Vorlesung und eine Übung statt.
6	**Lernziele und Kompetenzen**	Nach einem Besuch der Vorlesung Produktionssystematik sollen die Studierenden in der Lage sein: • Ziele, Strategien, Vision und Mission der Unternehmen beurteilen zu können; • sich in der Aufbau- und Ablauforganisation eines Unternehmen zurecht zu finden; • die Inhalte der wesentlichen Kernprozesse produzierender Unternehmen zu kennen; • die technische und administrative Auftragsabwicklung nachzuvollziehen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222 Nebenfach Maschinenbau: Fertigungsautomatisierung und Produktionstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 92880	**Robotics Frameworks** Robotics frameworks	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Robotics Frameworks (4 SWS) Übung: Exercise Robotics Frameworks (0 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Dr.-Ing. Alexander Kühl Dr.-Ing. Sebastian Reitelshöfer Nina Merz	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Jörg Franke
5	**Inhalt**	• Basic concepts of robotics • Basic concepts of the Robot Operating System • Simulation of robots in virtual environments • Computer vision and machine learning in the context of robotics • Path and gripping grasp planning • Localization, mapping and navigation of mobile robots • Flow control with state machines for complex robot tasks • Introduction to relevant software frameworks for specific tasks (Robot Operating System, Gazebo, OpenCV, Tensorflow) • Solving a complex practical task as a team
6	**Lernziele und Kompetenzen**	In this module, students independently implement advanced tasks in robotics and related topics such as simulation, computer vision and machine learning using concrete examples. In doing so, the students deal with various established software frameworks and learn how to use them. Students are taught the following technical and methodological competences: After completing the module, students will be able to • Classify important terms of robotics • Understand the challenges of modern robotics in relation to complex tasks and develop approaches to solve them. • Analyse and practically apply complex issues in robotics (robotics frameworks, simulation tools and frameworks for image processing and artificial intelligence) • Explain and apply methods of robot motion control and planning • Explain the self-localisation of mobile robots and examine it using examples The students additionally acquire and train the following personal and social competences within the framework of the team task: After completing the module, the students can • Independently solve preparatory tasks • Organize their working time • Work together with other students in a group in a goal-oriented manner • Assess their own strengths and use them in a targeted way in the team performance

7	**Voraussetzungen für die Teilnahme**	Prerequisites : Basic knowledge of programming languages C++ and Python, additional information can be found on StudOn
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 94940	**Technische Grundlagen des ressourcenschonenden und intelligenten Wohnens** Engineering fundamentals for resource-efficient and smart living	**2,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Technische Grundlagen des ressourcenschonenden und intelligenten Wohnens (vhb) (2 SWS)	2,5 ECTS
3	Lehrende	Felix Funk Prof. Dr.-Ing. Jörg Franke	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Jörg Franke
5	**Inhalt**	Ebenso wie die Sektoren Verkehr und Industrie, gerät auch das private Wohnen zunehmend in das Spannungsfeld aus Ressourcenschonung und demografischem Wandel. Mit intelligenter Automatisierungstechnik ist es möglich, diesen Herausforderungen zu begegnen. Eine besondere Beachtung ist hier den soziologischen und ökonomischen Bedarfen zu schenken. Folgende Themenschwerpunkte werden im Rahmen der virtuellen Vorlesung adressiert: • Energieerzeugung, -speicherung und -verteilung im privaten Umfeld • Energieeffizient Wohnen mit intelligenter Automatisierungstechnik • Steigerung von Sicherheit und Komfort durch nutzergerechte Hausautomation • Betrachtung soziologischer, technologischer und ökonomischer Begleitfaktoren
6	**Lernziele und Kompetenzen**	Nach Bearbeitung der Lehrveranstaltung sollen Sie als Studierende folgende Lernziele erreicht haben: • Der Begriff Smart Home und die Interdependenzen seiner Domänen sind Ihnen bekannt • Sie kennen die Charakteristiken der technischen Anlagen zur Stromerzeugung und deren physikalischen Grundlagen • Sie sind fähig je nach Anforderung ein geeignetes Heizsystem auszuwählen • Sie kennen die Grundlagen zu Transport- und Verteilung elektrischer Energie • Die Problematik der Anbindung dezentraler, regenerativer Erzeugungsanlagen an das elektrische Versorgungsnetz ist Ihnen bekannt • Ein Überblick zu vorhandener Sensorik und Aktorik im AAL-Bereich herrscht vor • Sie kennen die charakteristischen Vor- und Nachteile der verschiedenen etablierten Kommunikationstechnologien im Smart-Home-Umfeld • Sie können Prozesse und Methoden aufzählen und erklären, die für eine technische Realisierung eines sich selbst organisierenden Smart Homes wichtig sind

		• Sie haben einen Überblick gewonnen, wie die Geräteklassen zur Realisierung ganzheitlicher Anwendungsszenarien verknüpft werden können • Sie kennen die grundlegenden Begriffe aus dem Innovationsmanagement und der Innovationsforschung • Der Begriff Akzeptanz ist Ihnen in seinen unterschiedlichen Dimensionen bekannt
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Maschinenbau: Computerintegrierte Produktion Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 45 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Psychologie als Nebenfach für Informatik

1	**Modulbezeichnung** 78389	**Anwendungsmodul** Applied module	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Vorlesung - Klinische Psychologie I: PS - Psychische Störungen 2 (2 SWS) Vorlesung: Vorlesung - Klinische Psychologie II: PS - Prävention, Rehabilitation, Berufsethik, Berufsrecht (0 SWS) Vorlesung: Vorlesung Arbeitspsychologie (2 SWS) Vorlesung: Klinische Psychologie I: PS (Vorlesung) - Psychische Störungen 1 (2 SWS, WiSe 2023) Vorlesung: Klinische Psychologie II: PS (Vorlesung) - Verfahrenslehre (2 SWS, WiSe 2023) Vorlesung: Organisationspsychologie (2 SWS, WiSe 2023) Vorlesung: Vorlesung Pädagogische Psychologie (SWS)	3 ECTS - 5 ECTS 3 ECTS 3 ECTS - -
3	Lehrende	Prof. Dr. Matthias Berking Lydia Rupp Prof. Dr. Cornelia Niessen	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Psychologie als Nebenfach für Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	schriftlich oder mündlich schriftlich oder mündlich schriftlich oder mündlich schriftlich oder mündlich schriftlich oder mündlich mündlich
11	**Berechnung der Modulnote**	schriftlich oder mündlich (100%) schriftlich oder mündlich (100%) schriftlich oder mündlich (100%) schriftlich oder mündlich (100%) schriftlich oder mündlich (100%) mündlich (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 78386	**Einführungsmodul** Introductory module	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Vorlesung Einführung in die Psychologie (2 SWS, WiSe 2023)	2 ECTS
3	Lehrende	Prof. Dr. Tim Rohe	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Psychologie als Nebenfach für Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur
11	**Berechnung der Modulnote**	Klausur (0%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 78388	**Grundlagenmodul** Introductory module	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Vorlesung Sozialpsychologie, Teil 1 (2 SWS) Vorlesung: Vorlesung Differentielle und Persönlichkeitspsychologie Teil 1 (2 SWS, WiSe 2023) Proseminar: Proseminar Biologische Psychologie (2 SWS) Vorlesung: Vorlesung Entwicklungspsychologie, Teil 2 (2 SWS) Vorlesung: Vorlesung Allgemeine Psychologie I: Wahrnehmung (2 SWS, WiSe 2023) Vorlesung: Vorlesung Allgemeine Psychologie II, Teil 1 (2 SWS) Vorlesung: Vorlesung Sozialpsychologie, Teil 2 (2 SWS) Vorlesung: Vorlesung Biologische Psychologie (2 SWS, WiSe 2023) Proseminar: Proseminar Allgemeine Psychologie II (2 SWS) Proseminar: Proseminar Allgemeine Psychologie I: Kognition (2 SWS) Vorlesung: Vorlesung Entwicklungspsychologie, Teil 1 (2 SWS) Vorlesung: Vorlesung Entwicklungspsychologie Teil 1 und 2 (SWS, WiSe 2023)	3 ECTS 3 ECTS - 3 ECTS 3 ECTS 3 ECTS 3 ECTS - 3 ECTS - 3 ECTS -
3	Lehrende	Dr. Nicole Methner Prof. Dr. Susanne Bruckmüller Prof. Dr. Frieder Lang Prof. Dr. Nicolas Rohleder Prof.Dr.Dr. Gottfried Spangler Prof. Dr. Tim Rohe Prof. Dr. Oliver Schultheiss	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Psychologie als Nebenfach für Informatik Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	schriftlich oder mündlich schriftlich oder mündlich schriftlich oder mündlich schriftlich oder mündlich schriftlich oder mündlich schriftlich oder mündlich
11	**Berechnung der Modulnote**	schriftlich oder mündlich (100%) schriftlich oder mündlich (100%) schriftlich oder mündlich (100%) schriftlich oder mündlich (100%) schriftlich oder mündlich (100%) schriftlich oder mündlich (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Jonas, K, Stroebe, W., & Hewstone, M. (2014). Sozialpsychologie. Heidelberg: Springer. Weitere relevante Literatur wird in der Veranstaltung bekannt gegeben. Jonas, K, Stroebe, W., & Hewstone, M. (2014). Sozialpsychologie. Heidelberg: Springer. Weitere relevante Literatur wird in der Veranstaltung bekannt gegeben. Brandstätter, V., Schüler, J., Puca, R. M., & Lozo, L. (2. Aufl., 2018). Motivation und Emotion. Berlin: Springer. (Teil II: Emotion) Mazur, James E. (2006) Lernen und Verhalten (6., aktualisierte Auflage). München: Pearson Studium. (Hieraus: alle Kapitel außer 11, 13, 14) Brandstätter, V., Schüler, J., Puca, R. M., & Lozo, L. (2. Aufl., 2018). Motivation und Emotion. Berlin: Springer. (Teil II: Emotion) Mazur, James E. (2006) Lernen und Verhalten (6., aktualisierte Auflage). München: Pearson Studium. (Hieraus: alle Kapitel außer 11, 13, 14) Brandstätter, V., Schüler, J., Puca, R. M., & Lozo, L. (2013). Motivation und Emotion. Berlin: Springer.(Teil I: Motivation) Schultheiss, O. C., & Wirth, M. M. (in press). Biopsychological aspects of motivation. In J. Heckhausen (Ed.), Motivation and action (3 ed.). New York: Cambridge University Press. Brandstätter, V., Schüler, J., Puca, R. M., & Lozo, L. (2013). Motivation und Emotion. Berlin: Springer.(Teil I: Motivation)

		Schultheiss, O. C., & Wirth, M. M. (in press). Biopsychological aspects of motivation. In J. Heckhausen (Ed.), Motivation and action (3 ed.). New York: Cambridge University Press.

Praktikum Informatik

1	**Modulbezeichnung** 93094	**Applied Software Engineering Bachelor-Praktikum** Applied software engineering Bachelor’s Internship	**10 ECTS**
2	Lehrveranstaltungen	Praktikum: Applied Software Engineering Bachelor-Praktikum (SWS)	-
3	Lehrende	Prof. Dr. Dirk Riehle	

4	**Modulverantwortliche/r**	Julian Hirsch Prof. Dr. Dirk Riehle
5	**Inhalt**	• Einführung in die einzelnen Phasen der Softwareentwicklung: Anforderungsanalyse, Spezifikation, Entwurf, Implementierung, Test, Wartung • Beispielhafter Einsatz ausgewählter repräsentativer Verfahren zur Unterstützung dieser Entwicklungsphasen • Ergonomische Prinzipien Benutzungsoberfläche • Objektorientierte Analyse und Design mittels UML • Entwurfsmuster als konstruktive, wiederverwendbare Lösungsansätze für ganze Problemklassen • Automatisch unterstützte Implementierung aus UML-Diagrammen • Teststrategien • Refactoring zur Unterstützung der Wartungsphase
6	**Lernziele und Kompetenzen**	Die Studierenden • wenden auf Basis der bereits erworbenen Programmierkenntnisse systematische und strukturierte Vorgehensweisen (wie das Wasserfall- und V-Modell) zur Bewältigung der Komplexität im Zusammenhang mit dem Programmieren-im-Großen" an; • benutzen ausgewählte Spezifikationssprachen (wie Endliche Automaten, Petri-Netze und OCL), um komplexe Problemstellungen eindeutig zu formulieren und durch ausgewählte Entwurfsverfahren umzusetzen; • wenden UML-Diagramme (wie Use Case-, Klassen-, Sequenz- und Kommunikationsdiagramme) zum Zweck objektorientierter Analyse- und Design-Aktivitäten an; • reproduzieren allgemeine Entwurfslösungen wiederkehrender Probleme des Software Engineering durch Verwendung von Entwurfsmustern; • erfassen funktionale und strukturelle Testansätze; • setzen Refactoring-Strategien zur gezielten Erhöhung der Software-Änderungsfreundlichkeit um.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 3
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	
16	**Literaturhinweise**	Lehrbuch der Softwaretechnik (Band 1), Helmut Balzert, 2000

1	**Modulbezeichnung** 240715	**Grafik-Praktikum Game Programming** Laboratory course: Games programming (GraPra)	**10 ECTS**
2	Lehrveranstaltungen	Praktikum: GraPra Game Programming (10 SWS)	10 ECTS
3	Lehrende	Prof. Dr. Marc Stamminger Laura Fink Linus Franke	

4	**Modulverantwortliche/r**	Dr.-Ing. Frank Bauer Prof. Dr. Tim Weyrich
5	**Inhalt**	Wie entstehen eigentlich digitale Inhalte in einem Museum? In diesem Projekt-Seminar lernen Sie den zugrundeliegenden Ablauf in Zusammenarbeit mit dem Germanischen Nationalmuseum kennen. Dabei arbeiten Sie selbst an jedem Schritt mit: vom Scan im Museum über die 3D-Rekonstruktion und Aufbereitung der Daten (in Blender) bis zur inhaltlichen Gestaltung (mit Unity 3D) und finalen Veröffentlichung z.B. über Sketchfab. Durch das Semester werden Sie immer begleitet von Experten aus dem Germanischen Nationalmuseum, Mitarbeitern der Computergrafik und der Digital Humanities in Erlangen. So können Sie im Laufe des Semesters eine digitale Ausstellung von Anfang bis Ende interaktiv und spannend gestalten.
6	**Lernziele und Kompetenzen**	Fachkompetenz Erschaffen Lernende planen, entwerfen und produzieren unter Zuhilfenahme von typischen Werkzeugen der Computergrafik eine digitale Ausstellung für eine reales Museumsobjekt.
7	**Voraussetzungen für die Teilnahme**	Sie sollten bereits über grundlegende Programmierkenntnisse verfügen. Mögliche Vorlesungen sind z.B. AuD, IWGS oder GdI. Idealerweise haben Sie auch bereits Erfahrung im Umgang mit 3D-Software oder die Veranstaltung Computergrafik absolviert.
8	**Einpassung in Studienverlaufsplan**	Semester: 7
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 40 h Eigenstudium: 260 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93192	**Hackerpraktikum Bachelor** Hacking lab (Bachelor´s students)	**10 ECTS**
2	Lehrveranstaltungen	Praktikum: Hackerpraktikum (Bachelor + Master) (0 SWS)	10 ECTS
3	Lehrende	Julian Geus Prof. Dr.-Ing. Felix Freiling	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Felix Freiling
5	**Inhalt**	In diesem Praktikum lernen die Teilnehmer den kritischen Umgang mit offensiver IT-Sicherheit. Es werden prinzipielle Angriffskonzepte erörtert und in einer abgeschotteten Umgebung gezielt zur Anwendung gebracht. Durch diese praktischen Erfahrungen aus der Sichtweise eines "Hackers" werden die Teilnehmer bzgl. sicherheitsrelevanten Fragen sensibilisiert und können die gewonnenen Erkenntnisse dann letztendlich auch zur Absicherung von Systemen einsetzen. Das Hackerpraktikum wird in bis zu 6 Übungsblätter zu je 3 Wochen aufgeteilt, wobei die folgenden beispielhaften Themen bearbeitet werden: • Netzwerksicherheit (Sniffing, Spoofing, WPA, ...) • Webhacking (SQL Injections, XSS, CSRF, ...) • Systemsicherheit (Rootkits, Privileges, Suid, ...) • Reverse Engineering (Cracking, Malware Analysis, ...) • Exploitation (Buffer Overflows, Shellcode, ASLR, ...) Von den Übungen müssen insgesamt 2/3 und pro Blatt 1/2 der möglichen Punkte erreicht werden. Die Übungen sind in Einzelarbeit abzugeben.
6	**Lernziele und Kompetenzen**	Die Studierenden können Schwachstellen in den vorgestellten Themenbereichen identifizieren und beschreiben. Sie können aktuelle Angriffs- und Verteidigungstechniken in konkreten Fällen auswählen und anwenden.
7	**Voraussetzungen für die Teilnahme**	• Interesse an IT-Sicherheit. • Bereitschaft, viel Zeit in das Praktikum zu investieren. • Progammierkenntnisse in C/Assembler und mindestens einer Skriptsprache (bspw. Python). • Linux-Kenntnisse. • Von großem Vorteil sind darüber hinaus Netzwerk-Kenntnisse und Vorkenntnisse im Bereich IT-Sicherheit.
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	nur im Wintersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 50 h Eigenstudium: 250 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 695344	**HPC Software Projekt** HPC Software project	**10 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Harald Köstler
5	**Inhalt**	Anhand eines aktuellen Forschungsthema im Bereich High Performance Computing sollen die Studierenden an die wissenschaftliche Arbeitsweise im Bereich Informatik herangeführt werden. Dazu wird typischerweise in Gruppenarbeit ein größeres Softwarepaket entwickelt und auf eine konkrete Problemstellung aus dem HPC Bereich angewendet. Beispiele sind die Implementierung und Parallelisierung eines Simulationscodes für Anwendungsprobleme aus der Strömungsmechanik oder eines neuronalen Netzes für Anwendungsprobleme, die mit Hilfe von künstlicher Intelligenz gelöst werden können.
6	**Lernziele und Kompetenzen**	Fachkompetenz Verstehen Lernende können in Veröffentlichungen beschriebene oder in bestehender Software enthaltene numerische Algorithmen verstehen. Anwenden Lernende können numerische Algorithmen auf vorgegebene Problemstellungen aus dem Bereich High Performance Computing (HPC) anwenden. Analysieren Auswahl von geeigneten numerischen Algorithmen und effizienten, parallelen Datenstrukturen, um ein vorgegebenes Problem auf einer vorgegebenen Hardware-Plattform effizient zu lösen. Erschaffen Entwicklung eines Softwarepaketes für eine konkrete Problemstellung aus dem High Performance Computing (HPC) Bereich. Lern- bzw. Methodenkompetenz Die Studierenden erwerben die Kompetenz, eigenständig Software zu erstellen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 7
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung

		Zur Bewertung der Praktikumsleistung wird das Material aus der Abgabe der Software- und Datenartefakte (90% Gewichtung) und der abschliessenden Ergebnispräsentation (Vortrag 15 min, 10% Gewichtung) herangezogen.
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 25 h Eigenstudium: 275 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93199	**IoT Security** IoT security	**10 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Felix Freiling Philipp Klein
5	**Inhalt**	Dieses Projekt wird in kleinen Teams bearbeitet. Jedes Team muss am Ende einen schriftlichen Bericht einreichen. Die Arbeit der einzelnen Teammitglieder muss hier klar ersichtlich sein. Jedes Team erhaelt eine Reihe von IoT- und Smart-Home-Geraeten. Diese sollen zunaechst zu einem funktionierenden, "smarten" System verknuepft werden. Es werden eigene Geraete gebaut und mit dem System verbunden. Im Anschluss wird detailliert fuer jedes Geraet eine Sicherheitsanalyse durchgefuehrt. Hier wird unter anderem betrachtet, welche Daten die Geraete wohin senden, welche Daten lokal gespeichert und erhoben werden und wie die Kommunikation abgesichert ist. Der schriftliche Bericht beinhaltet die Beschreibung und Dokumentation des Systems und der selbstgebauten IoT-Geraete sowie die Erkenntnisse der Sicherheitsanalyse. Zusammengefasster Inhalt: • Grundlagenvorlesungen zu IoT und Smart Home • Verknuepfung von diversen IoT-Geraeten • Bau eigener IoT-Geraete • Detaillierte Analyse von IoT-Geraeten bezueglich Datenverkehr, Datenschutz und Usability • Ausarbeitung und Anwendung von Angriffsszenarien auf IoT-Geraete
6	**Lernziele und Kompetenzen**	Fachkompetenz Evaluieren (Beurteilen) Lernende koennen aufgrund Ihrer Erfahrung im Projekt bewerten, ob ein IoT-Geraet als "sicher" einzustufen ist. Diese Bewertung erfolgt entweder auf Grundlage von frei verfuegbaren Daten oder ueber eine selbststaendige Evaluation des Geraets. Selbstkompetenz Lernende koennen eigenstaendig, ohne Aufsicht und Anleitung, ein IoT-Geraet detailliert analysieren und zu festgelegten Zeitpunkten Ergebnisse vorweisen.

		Sozialkompetenz Lernende koennen in kleinen Teams effektiv und effizient zusammenarbeiten, die Arbeit gerecht verteilen und gemeinsam einen Bericht anfertigen.
7	**Voraussetzungen für die Teilnahme**	Grundkenntnisse in Python
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%) Die Modulnote setzt sich aus der individuellen Arbeit, einem abschliessenden Report und dem Abschlussvortrag zusammen. Die Gewichtung liegt bei 50:40:10.
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 20 h Eigenstudium: 280 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93203	**Mobile Application Development and Security** Mobile application development and security	**10 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Philipp Klein
5	**Inhalt**	In diesem Projekt/Praktikum erlernt ihr anhand eines Vorlesungsteils und eines anschliessenden Team-Projekts, mobile Applikationen zu Erstellen (Backend und App) und auszuliefern (Deployment, DevOps). Inhalte sind u.a.: • Entwicklung von Android-Applikationen mit Kotlin • Entwicklung von Cross-Platform-Apps mit Flutter • Continuous Integration mit GitlabCI • Agile Entwicklung im Team, angepasst an die Bedürfnisse des Teams • Statische und dynamische Analyse von Apps • Sichere Entwicklung im Open-Source-Umfeld In diesem Praktikum/Projekt agieren wir wie ein Software-Unternehmen. Es wird einen Auftraggeber, ein Lasten- und Pflichtenheft sowie Meilensteine und Deadlines geben. Die Arbeit geschieht in einem Team mit einem woechentlichen Meeting. In der initialen Lernphase bekommt ihr Vorlesungsmaterial und Uebungen zur Verfuegung gestellt. In der anschliessenden Projektphase wird an einer App gearbeitet. Aktuell ist dies die offizielle FAU-App, die neu entwickelt wird.
6	**Lernziele und Kompetenzen**	Fachkompetenz Erschaffen Studierende planen und entwickeln Android- und iOS-Applikationen. Sie legen in der Konzeption und Umsetzung grossen Wert auf Sicherheit. Lern- bzw. Methodenkompetenz Studierende wenden sämtliche Arbeitsschritte in der Entwicklung mobiler Applikationen, von der Planung über die Erstellung einer CI-Pipeline bis zur Auslieferung, an. Selbstkompetenz Studierende integrieren sich in Teams, planen das gemeinsame Vorgehen und reflektieren über das bisher Geleistete. Sozialkompetenz Studierende dokumentieren ihre Arbeit nach aussen in Form eines Blogs auch für nicht-technische Studierende verständlich und nachvollziehbar. Sie kommunizieren innerhalb des Teams respektvoll und wertschätzend.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!

9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%) Die Note berechnet sich aus der Arbeit im Team sowie einem Abschlussvortrag. Die Gewichtung beträgt 90:10. Bei der Teamarbeit wird sowohl der geschriebene Code, als auch das Verhalten im Team bewertet.
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 917928	**Nailing your Thesis (VUE+PROJ 10-ECTS)** Nailing your thesis (VUE+PROJ 10-ECTS)	**10 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Dirk Riehle
5	**Inhalt**	This course teaches students how to perform scientific research for their final thesis or a research paper. The goal is to prepare students for a Bachelor or Master research thesis. The course covers the following topics: • Science and society • The research process • Theory building research • Theory validation research • Writing a research thesis/paper • The scientific community Students can choose one or both of two components: • VUE (VL + UE or seminar), 4 SWS, 5 ECTS. VUE combines lectures with homework and exercises. VUE is run as a 3h block. • PROJ (small research project), 2 SWS, 5 ECTS. In PROJ, students perform a small research project, either individually or in teams. The available projects will be presented at the beginning of the course. Students perform the research, write a paper, and hold a presentation about their work. Sign-up and further course information are available at https://nyt.uni1.de - please sign up for the course on StudOn (available through previous link) as soon as possible. The course information will also tell you how the course will be held (online or in person).
6	**Lernziele und Kompetenzen**	• Students gain an understanding of how science works • Students learn how to perform research work • Students learn how to write a research thesis
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung

11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	See https://nyt.uni1.de

1	**Modulbezeichnung** 93129	**NWERC Praktikum** NWERC Training	**10 ECTS**
2	Lehrveranstaltungen	Praktikum: NWERC Praktikum (8 SWS)	10 ECTS
3	Lehrende	Prof. Dr. Michael Philippsen Daniela Novac Paul Wild	

4	**Modulverantwortliche/r**	Prof. Dr. Michael Philippsen
5	**Inhalt**	Programmierwettbewerbe wie der International Collegiate Programming Contest (ICPC) bieten eine Möglichkeit, die eigenen Programmier- und Teamfähigkeiten an einer Vielzahl algorithmischer Probleme zu testen. Dabei stammen die Aufgaben aus verschiedenen Gebieten, wie Geometrie, Kombinatorik, String-Verarbeitung und Zahlentheorie. Die Studierenden treten in 3er-Teams an, haben aber nur einen Computer zur Verfügung. Neben fachlichem Verständnis ist die Teamstrategie entscheidend für den Erfolg der Gruppe. Die Wettbewerbe werden auf drei Ebenen abgehalten: • Lokal an jeder Universität. In Deutschland findet diese in der Regel im Sommer unter dem Namen "German Collegiate Programming Contest - GCPC" statt. An diesem Wettbewerb können alle Studierenden teilnehmen. • Regional in weltweit mehr als 30 Regionalausscheidungen, zu denen jede Universität maximal 3 Teams entsenden darf. Die FAU nimmt am North Western European Regional Contest (NWERC) teil, bei dem jeweils im Herbst die besten Teams für das Finale ermittelt werden. • Die World Finals finden im Frühling des darauffolgenden Jahres statt. Die zwei besten Teams jeder Region dürfen an den World Finals teilnehmen. Dieses Praktikum richtet sich an Studierende, die bereits am lokalen Wettbewerb (GCPC) teilgenommen haben und sich dabei durch herausragende Leistung für die engere Auswahl qualifiziert haben, um für die FAU beim NWERC antreten zu dürfen. Das Praktikum setzt sich aus drei Komponenten zusammen: • Teamcontests (Juni bis Mitte November): Über die gesamte Praktikumsdauer werden voraussichtlich 25 Probe-Wettbewerbe (die genaue Anzahl kann erst zu Semesterbeginn festgelegt werden), zu je 5 Stunden abgehalten (insgesamt ca. 200 Aufgaben). Die Studierenden trainieren dabei in 3er-Teams mit wechselnder Besetzung. Nach jedem Probe-Wettbewerb werden die Lösungsansätze der gestellten Aufgaben besprochen. Danach haben die Teilnehmer zwei Wochen lang die Möglichkeit, eigenständig erarbeitete Lösungen für ungelöste Aufgaben auf der zur Verfügung gestellten Plattform (DOMJudge) einzureichen. Eine Aufgabe wird als gültig eingestuft, wenn sie alle Testfälle in einer vorgegebenen Zeit bestanden hat. Alle eingereichten Lösungsversuche werden sofort und automatisch evaluiert.

		Die Studierenden erhalten eine entsprechende Rückmeldung und es dürfen beliebig viele Lösungsversuche eingereicht werden. • Trainingslager (Anfang September): Anfang September findet ein 3-tägiges Trainingslager statt, bei dem anhand von weiteren Probe-Wettbewerben Teamstrategien trainiert werden. Die Übernachtungskosten werden von der Universität übernommen. Die Teilnehmer müssen nur für die Verpflegung aufkommen. Die Teilname am Trainingslager ist verpflichtend, um die Bewertung der Teamfähigkeit zu ermöglichen. • Einzelcontests (bis Mitte September): Nach dem Trainigslager finden zwei 5-stündige Einzelwettbewerbe statt, an denen die Studierenden individuell teilnehmen und bewertet werden. Wie bei den Teamcontests werden die Aufgaben nachbesprochen und die Studierenden haben die Möglichkeit, Lösungen innerhalb von zwei Wochen nachzureichen. Die Leistung und Teamfähigkeit entscheiden über die Teilnahmemöglichkeit am Wettbewerb auf regionaler Ebene (NWERC). Bis Mitte September, nach den Einzelcontests, werden die neun best geeigneten Studierenden ausgewählt. Diese dürfen in drei Teams am NWERC für die FAU teilnehmen. Die Teilnahme und Platzierung am NWERC beeinflusst die Benotung nicht. Die Unterrichtssprache ist Deutsch, nur die Aufgabenstellungen sind in englischer Sprache verfasst.
6	**Lernziele und Kompetenzen**	*A - Methodenkompetenz* Die Studierenden • bewerten verschiedene Algorithmen hinsichtlich ihrer Eignung für vorgegebene Problemstellungen • priorisieren die Bearbeitung verschiedener Aufgaben, indem sie den Schwierigkeitsgrad und Umfang der zu erwartenden Lösung bewerten • entwickeln eine Strategie, die benötigten Algorithmen in kurzer Zeit zu implementieren und anzupassen • erarbeiten neue effiziente Algorithmen zur Lösung der Aufgaben und setzen diese schnell und fehlerfrei um • testen/überprüfen eigenen und fremden Code und beheben selbständig dabei gefundene Fehler *B - Selbst- und Sozialkompetenz:* Die Studierenden • kennen die eigenen Stärken und Schwächen und setzen dieses Wissen bei der Auswahl und Priorisierung der zu bearbeitenden Aufgaben zielführend ein • entwickeln gemeinsam eine Lösung(sstrategie) und unterstützen sich gegenseitig bei der Umsetzung • arbeiten in Gruppen kooperativ und verantwortlich, halten sich an vereinbarte Regeln und gehen offen auf andere zu • zeigen eine positive Grundhaltung anderen gegenüber, handeln partner- und situationsgerecht

		• gehen mit Konflikten angemessen um und kommunizieren und handeln fair
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 7
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung Das Praktikum setzt sich aus drei Komponenten zusammen: • Teamcontests (Juni bis Mitte November): Über die gesamte Praktikumsdauer werden voraussichtlich 25 Probe-Wettbewerbe (die genaue Anzahl kann erst zu Semesterbeginn festgelegt werden), zu je 5 Stunden abgehalten (insgesamt ca. 200 Aufgaben). Die Studierenden trainieren dabei in 3er-Teams mit wechselnder Besetzung. Nach jedem Probe-Wettbewerb werden die Lösungsansätze der gestellten Aufgaben besprochen. Danach haben die Teilnehmer zwei Wochen lang die Möglichkeit, eigenständig erarbeitete Lösungen für ungelöste Aufgaben auf der zur Verfügung gestellten Plattform (DOMJudge) einzureichen. Eine Aufgabe wird als gültig eingestuft, wenn sie alle Testfälle in einer vorgegebenen Zeit bestanden hat. Alle eingereichten Lösungsversuche werden sofort und automatisch evaluiert. Die Studierenden erhalten eine entsprechende Rückmeldung und es dürfen beliebig viele Lösungsversuche eingereicht werden. • Trainingslager (Anfang September): Anfang September findet ein 3-tägiges Trainingslager statt, bei dem anhand von weiteren Probe-Wettbewerben Teamstrategien trainiert werden. Die Übernachtungskosten werden von der Universität übernommen. Die Teilnehmer müssen nur für die Verpflegung aufkommen. Die Teilname am Trainingslager ist verpflichtend, um die Bewertung der Teamfähigkeit zu ermöglichen. • Einzelcontests (bis Mitte September): Nach dem Trainigslager finden zwei 5-stündige Einzelwettbewerbe statt, an denen die Studierenden individuell teilnehmen und bewertet werden. Wie bei den Teamcontests werden die Aufgaben nachbesprochen und die Studierenden haben die Möglichkeit, Lösungen innerhalb von zwei Wochen nachzureichen.
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 210 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	https://link.springer.com/book/10.1007/978-3-319-72547-5

1	**Modulbezeichnung** 93142	**The AMOS Project (SD Role, PROJ 10 ECTS)** The AMOS Project (SD Role, Proj 10 ECTS)	**10 ECTS**
2	Lehrveranstaltungen	Übung: The AMOS Project (UE) (2 SWS) Vorlesung: The AMOS Project (VL) (2 SWS)	- -
3	Lehrende	Prof. Dr. Dirk Riehle	

4	**Modulverantwortliche/r**	Prof. Dr. Dirk Riehle
5	**Inhalt**	This course teaches agile methods (Scrum and XP) and open source tools using a single semester-long project. Topics covered are: • Agile methods and related software development processes • Scrum roles, process practices, including product and engineering management • Technical practices like refactoring, continuous integration, and test-driven development • Principles and best practices of open source software development The project is a software development project in which each student team works with an industry partner who provides the idea for the project. This is a practical hands-on experience. Students can play one of two primary roles: • Product owner. In this function, a student defines, prioritizes, communicates, and reviews requirements. The total effort adds up to 5 ECTS. • Software developer. In this function, a student estimates their effort for requirements and implements them. The total effort adds up to 10 ECTS. Students will be organized into teams of 7-8 people, combining product owners with software developers. An industry partner will provide requirements to be worked out in detail by the product owners and to be realized by the software developers. The available projects will be presented in the run-up to the course. Class consists of a 90min lecture followed by a 90min team meeting. Rooms and times for team meetings are assigned at the beginning of the semester. You must be able to regularly participate in the team meetings. If you can't, do not sign up for this course. Students choosing the software developer role must have prior software development experience. Sign-up and further course information are available at https://amos.uni1.de - please sign up for the course on StudOn (available through previous link) as soon as possible. The course information will also tell you how the course will be held (online or in person).
6	**Lernziele und Kompetenzen**	• Students learn about software products and software development in an industry context • Students learn about agile methods, in particular, Scrum and Extreme Programming

		• Students gain practical hands-on experience with a Scrum process and XP technical practices
7	**Voraussetzungen für die Teilnahme**	- For software developer role: OSS-ADAP
8	**Einpassung in Studienverlaufsplan**	Semester: 7
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 113845	**Praktikum angewandte Systemsoftwaretechnik** Laboratory course: Applied systems software technology	**10 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 594684	**Praktikum Enterprise Computing**	**10 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93205	**Praktikum Entwicklung digitaler Lernstationen** Laboratory course: Development of digital learning stations	**10 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Prof. Dr. Marc-Pascal Berges
5	**Inhalt**	Die Studierenden planen eigenständig in Kleingruppen digitale Lernstationen. Dabei steht das explorative Lernparadigma im Vordergrund. User Experience, Motivation und Selbstwirksamkeitserwartungen werden thematisiert und von den Studierenden in den Lernstationen umgesetzt. Agile Softwareentwicklung im Team bildet die Grundlage für die Entwicklung der Lernstationen.
6	**Lernziele und Kompetenzen**	Die Studierenden sind in der Lage eine digitale Lernstation unter Berücksichtigung von Lerntheorien zu planen. Die Studierenden sind in der Lage sich als Team auf Basis agiler Methoden zu organisieren. Die Studierenden sind in der Lage eine digitale Lernstation zu implementieren.
7	**Voraussetzungen für die Teilnahme**	Grundlegende Programmierkenntnisse
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung Die Studierenden erstellen einen Praktikumsbericht, der sowohl den Lernprozess im Sinne des refexiven Schreibens, als auch die entstandenen Produkte dokumentiert.
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 716025	**Praktikum: Entwicklung interaktiver eingebetteter Systeme** Laboratory course: Development of cyber-physical embedded systems	**10 ECTS**
2	Lehrveranstaltungen	Praktikum: Praktikum: Entwicklung interaktiver eingebetteter Systeme (8 SWS)	10 ECTS
3	Lehrende	Joachim Falk	

4	**Modulverantwortliche/r**	Joachim Falk
5	**Inhalt**	Technische Systeme machen den Alltag immer angenehmer. Unsere Ansprüche steigen dabei stetig und so erwarten wir, dass unsere Geräte immer mehr Funktionen bieten, und gleichzeitig immer einfacher zu bedienen sind. Eingebettete Systeme, also spezialisierte Computer, die direkt in technische Systeme integriert sind, sind dabei seit Jahren das Mittel der Wahl, um unseren Geräten die benötigte Intelligenz zu verpassen. Es hat sich gezeigt, dass es immer wichtiger wird, dass diese Systeme noch mehr mit ihrer Umwelt und vor allem dem Menschen interagieren. Ein Paradebeispiel hierfür sind Fahrerassistenzsysteme in modernen Fahrzeugen. Derartige Systeme erhöhen nicht nur den Komfort, sondern vor allem auch die Sicherheit aktueller Automobile. Diese stark mit ihrer Umwelt interagierenden, eingebetteten Systeme nennt man auch Cyberphysical Systems. Das Praktikum Entwicklung interaktiver eingebetteter Systeme" behandelt die Entwicklung und Programmierung derartiger Systeme. In drei Abschnitten bietet das Praktikum Einblick in alle Phasen der Entwicklung eingebetteter Systeme. Dabei entwickeln wir ein Objekterkennungssystem, bei dem ein Objekt erkannt werden soll, um dessen Position zur Steuerung der von Ihnen entwickelten Software einzusetzen. • In Phase I entwickeln wir Filter- und Objekterkennungsalgorithmen, mit deren Hilfe Objekte automatisch in einem Videostrom detektiert werden sollen. Das Erkennen soll möglichst zuverlässig und bei wechselnden Umgebungsbedingungen funktionieren. • In Phase II portieren wir die entwickelten Algorithmen auf das eingebettete System. Wir verändern allerdings nicht die Programmierung eines Autos, sondern führen die Umsetzung anhand eines virtuellen Prototypen durch. Mittels des virtuellen Prototypen entscheiden wir dann, was in Software und was in Hardware implementiert werden soll. • In Phase III testen wir unsere Entwicklungen auf einem realen System. Für diesen Zweck steht uns ein am Lehrstuhl entwickelter Demonstrator zur Verfügung. Dadurch lassen sich die entwickelten Filter auf einem realen System testen. Der Demonstrator verfügt über eine Kamera, die den Bereich vor dem Demonstrator überwacht. Die Kameradaten werden anschließend an die selbst entwickelten Module weitergeleitet.

		Diese erzeugen im Anschluss Steuersignale für einen Motor, der einen verfahrbaren Schlitten antreibt.
6	**Lernziele und Kompetenzen**	Fachkompetenz - Wissen • Die Studierenden legen Konzepte des modellbasierten Entwurfs eingebetteter Systeme dar. Fachkompetenz - Verstehen • Die Studierenden veranschaulichen die Hauptaufgaben beim Systementwurf auf verschiedenen Abstraktionsebenen von der Anwendung selbst, dem Gesamtsystem, bis hinunter zu einem Hardwaremodul. • Die Studierenden schildern den Entwurf eines System von der Idee, über die Spezifikation bis zur Implementierung, der Analyse und letztendlich der Validierung an einem realen mechatronischen Versuchsaufbau. Fachkompetenz - Anwenden • Die Studierenden setzen die Integration von digitalen Hardware- und Software-Komponenten um. • Die Studierenden wenden die Programmiersprachen C/C++/ SystemC für die Entwicklung von Hardware- und Software-Komponenten an. Selbstkompetenz • Die Studierenden schätzen ihre individuellen Stärken ab, um einen geeignete Arbeitsaufteilung innerhalb der Gruppe festzulegen. Sozialkompetenz • Die Studierenden erarbeiten gemeinsam Schnittstellendefinitionen über Gruppengrenzen hinweg. • Die Studierenden erarbeiten kooperativ in Gruppen Lösungskonzepte und implementieren diese gemeinsam.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung Die Sprache des Abschlussvortrags ist abhängig von der Wahl des Studierenden entweder Deutsch oder Englisch.
11	**Berechnung der Modulnote**	Praktikumsleistung (100%) Die Bewertung der Prüfungsleistung setzt sich zusammen aus Abschlussvortrag (Dauer: 25 min) und erfolgreicher Bearbeitung aller Praktikumsaufgaben (verpflichtend). Die Modulnote ergibt sich aus 50% Abschlussvortrag und 50% Mitarbeit im Praktikum.
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 120 h Eigenstudium: 180 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/praktika/entwicklung-interaktiver-eingebetteter-systeme/

1	**Modulbezeichnung** 93197	**Praktikum: Entwicklung interaktiver eingebetteter Systeme** Laboratory course: Development of cyber-physical embedded systems	**10 ECTS**
2	Lehrveranstaltungen	Praktikum: Praktikum: Entwicklung interaktiver eingebetteter Systeme (8 SWS)	10 ECTS
3	Lehrende	Joachim Falk	

4	**Modulverantwortliche/r**	Joachim Falk
5	**Inhalt**	Technische Systeme machen den Alltag immer angenehmer. Unsere Ansprüche steigen dabei stetig und so erwarten wir, dass unsere Geräte immer mehr Funktionen bieten, und gleichzeitig immer einfacher zu bedienen sind. Eingebettete Systeme, also spezialisierte Computer, die direkt in technische Systeme integriert sind, sind dabei seit Jahren das Mittel der Wahl, um unseren Geräten die benötigte Intelligenz zu verpassen. Es hat sich gezeigt, dass es immer wichtiger wird, dass diese Systeme noch mehr mit ihrer Umwelt und vor allem dem Menschen interagieren. Ein Paradebeispiel hierfür sind Fahrerassistenzsysteme in modernen Fahrzeugen. Derartige Systeme erhöhen nicht nur den Komfort, sondern vor allem auch die Sicherheit aktueller Automobile. Diese stark mit ihrer Umwelt interagierenden, eingebetteten Systeme nennt man auch Cyberphysical Systems. Das Praktikum Entwicklung interaktiver eingebetteter Systeme" behandelt die Entwicklung und Programmierung derartiger Systeme. In drei Abschnitten bietet das Praktikum Einblick in alle Phasen der Entwicklung eingebetteter Systeme. Dabei entwickeln wir ein Objekterkennungssystem, bei dem ein Objekt erkannt werden soll, um dessen Position zur Steuerung der von Ihnen entwickelten Software einzusetzen. • In Phase I entwickeln wir Filter- und Objekterkennungsalgorithmen, mit deren Hilfe Objekte automatisch in einem Videostrom detektiert werden sollen. Das Erkennen soll möglichst zuverlässig und bei wechselnden Umgebungsbedingungen funktionieren. • In Phase II portieren wir die entwickelten Algorithmen auf das eingebettete System. Wir verändern allerdings nicht die Programmierung eines Autos, sondern führen die Umsetzung anhand eines virtuellen Prototypen durch. Mittels des virtuellen Prototypen entscheiden wir dann, was in Software und was in Hardware implementiert werden soll. • In Phase III testen wir unsere Entwicklungen auf einem realen System. Für diesen Zweck steht uns ein am Lehrstuhl entwickelter Demonstrator zur Verfügung. Dadurch lassen sich die entwickelten Filter auf einem realen System testen. Der Demonstrator verfügt über eine Kamera, die den Bereich vor dem Demonstrator überwacht. Die Kameradaten werden anschließend an die selbst entwickelten Module weitergeleitet.

		Diese erzeugen im Anschluss Steuersignale für einen Motor, der einen verfahrbaren Schlitten antreibt.
6	**Lernziele und Kompetenzen**	Fachkompetenz - Wissen • Die Studierenden legen Konzepte des modellbasierten Entwurfs eingebetteter Systeme dar. Fachkompetenz - Verstehen • Die Studierenden veranschaulichen die Hauptaufgaben beim Systementwurf auf verschiedenen Abstraktionsebenen von der Anwendung selbst, dem Gesamtsystem, bis hinunter zu einem Hardwaremodul. • Die Studierenden schildern den Entwurf eines System von der Idee, über die Spezifikation bis zur Implementierung, der Analyse und letztendlich der Validierung an einem realen mechatronischen Versuchsaufbau. Fachkompetenz - Anwenden • Die Studierenden setzen die Integration von digitalen Hardware- und Software-Komponenten um. • Die Studierenden wenden die Programmiersprachen C/C++/ SystemC für die Entwicklung von Hardware- und Software-Komponenten an. Selbstkompetenz • Die Studierenden schätzen ihre individuellen Stärken ab, um einen geeignete Arbeitsaufteilung innerhalb der Gruppe festzulegen. Sozialkompetenz • Die Studierenden erarbeiten gemeinsam Schnittstellendefinitionen über Gruppengrenzen hinweg. • Die Studierenden erarbeiten kooperativ in Gruppen Lösungskonzepte und implementieren diese gemeinsam.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 7
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%) Die Bewertung der Prüfungsleistung setzt sich zusammen aus Abschlussvortrag (Dauer: 25 min) und erfolgreicher Bearbeitung aller wöchentlichen Praktikumsaufgaben (verpflichtend, vor Ort an den Rechnerarbeitsplätzen des Lehrstuhls). Die Modulnote ergibt sich aus 50% Abschlussvortrag und 50% Mitarbeit im Praktikum basierend auf den abgegebenen Praktikumsaufgaben.
12	**Turnus des Angebots**	nur im Wintersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 120 h Eigenstudium: 180 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/praktika/entwicklung-interaktiver-eingebetteter-systeme/

1	**Modulbezeichnung** 93187	**Praktikum Künstliche Intelligenz (BSc)** Laboratory course: Artificial intelligence (BSc)	**10 ECTS**
2	Lehrveranstaltungen	Praktikum: Praktikum Künstliche Intelligenz (BSc) (0 SWS)	-
3	Lehrende	Prof. Dr. Michael Kohlhase	

4	**Modulverantwortliche/r**	Prof. Dr. Michael Kohlhase
5	**Inhalt**	The KWARC group (Wissensrepräsentation und Verarbeitung) conducts research in knowledge representation and reasoning techniques with a view towards applications in knowledge management. We extend techniques from formal methods so that they can be used in settings where formalization is either infeasible or too costly. We concentrate on developing techniques for marking up the structural semantics in technical documents. This level of markup allows for offering interesting knowledge management services without forcing the author to formalize the document contents. In contrast to courses with fixed topics, project topics are defined individually. See http://kwarc.info for further information.
6	**Lernziele und Kompetenzen**	Die Studierenden • arbeiten sich in komplexe Softwaresysteme ein und erweitern diese • lernen, eigenständig Lösungsvorschläge auszuarbeiten und umzusetzen • dokumentieren die von ihnen geschriebene Software
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 278855	**Praktikum Lego Mindstorms** Laboratory course: Lego mindstorms	**10 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Joachim Falk Dr.-Ing. Stefan Wildermann
5	**Inhalt**	Informatik befasst sich nicht nur mit der Programmierung von Desktop-Rechnern. Vielmehr können Computer in immer mehr Gegenständen unseres Alltags oder in bestimmten technischen Kontexten gefunden werden. Man spricht hier von eingebetteten Systemen. Auch Roboter stellen solche eingebetteten Systeme dar. Ein Roboter erwacht durch sein Programm zum Leben. Die Programmierung von Robotern stellt einerseits eine Herausforderung dar. Andererseits ist sie aber auch mit viel Spaß verbunden. In diesem Modul werden LEGO Mindstorms Roboter verwendet, die mittels der Sprache Java programmiert werden. Dazu wird das Betriebssystem leJOS verwendet. Ziel des Moduls ist es, die Teamfähigkeit weiterzuentwickeln, indem ein praktisches Thema als Gruppe bearbeitet wird, Kenntnisse und Fertigkeiten in der Projektorganisation zu erwerben und die Fähigkeit der Problemlösung zu schulen.
6	**Lernziele und Kompetenzen**	**Fachkompetenz - Erschaffen** • Die Studierenden erstellen Lösungsideen für die Projekte und implementieren diese in Java für Lego Mindstorms Roboter. **Selbstkompetenz** • Die Studierenden schätzen ihre Stärken ab, um eine geeignete Arbeitsaufteilung innerhalb der Gruppe zu finden. **Sozialkompetenz** • Die Studierenden erarbeiten gemeinsam Projektpläne und -dokumentation im Themengebiet Robotik. • Die Studierenden organisieren selbstständig die gemeinsame Durchführung des Projekts und führen diese kooperativ in Gruppen durch.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%) Die Bewertung der Prüfungsleistung setzt sich zusammen aus erfolgreicher Umsetzung einer Projektaufgabe, einem Abschlussvortrag (Dauer: ca. 25 min) und einem Praktikumsbericht (ca. 12

		Seiten). Die Modulnote ergibt sich aus 20% Abschlussvortrag, 40% Praktikumsbericht und 40% Bewertung der umgesetzten Projektaufgabe.
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/praktika/lego-mindstorms-praktikum/

1	**Modulbezeichnung** 93155	**Praktikum Mustererkennung** Laboratory course: Pattern recognition	**10 ECTS**
2	Lehrveranstaltungen	Praktikum: Praktikum Mustererkennung (4 SWS)	5 ECTS
3	Lehrende	Dr.-Ing. Vincent Christlein	

4	**Modulverantwortliche/r**	Dr.-Ing. Vincent Christlein Felix Denzinger Fabian Wagner
5	**Inhalt**	At the Pattern Recognition Lab we offer practical topics that are connected to our current research in the fields of pattern recognition & machine learning, medical image processing, and big data applications. Other than a course with fixed topic, topics are defined individually. The 10 ECTS project is directed towards students of computer science. Also smaller 5 ECTS projects are available. The goal of this practical course is to familiarize the students with a Pattern Recognition System. Am Lehrstuhl für Mustererkennung bieten wir Praktika an, die mit unserer aktuellen Forschung in den Bereichen Mustererkennung & maschinelles Lernen, medizinische Bildverarbeitung und große Datenanwendungen verbunden sind. Im Gegensatz zu einem Kurs mit festem Thema werden die Themen individuell festgelegt. Das 10 ECTS-Projekt richtet sich an Studierende der Informatik. Auch kleinere 5 ECTS-Projekte sind möglich. Das Ziel dieses Praktikums ist es, die Studenten mit einem Mustererkennungssystem vertraut zu machen.
6	**Lernziele und Kompetenzen**	Die Studierenden • arbeiten sich in komplexe Softwaresysteme ein und erweitern diese • lernen, wie ein Mustererkennungssystem zu entwickeln ist • lernen, eigenständig Lösungsvorschläge auszuarbeiten und umzusetzen • dokumentieren die von ihnen geschriebene Software The students • familiarize themselves with complex software systems and extend them • learn to develop a pattern recognition system • learn to independently develop and implement proposed solutions • create documentation for the software written by them
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 7
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung At the Pattern Recognition Lab we offer practical topics that are connected to our current research in the fields of pattern recognition &

		machine learning, medical image processing, and big data applications. Other than a course with fixed topic, topics are defined individually. The 10 ECTS project is directed towards students of computer science. Also smaller 5 ECTS projects are available. The goal of this practical course is to familiarize the students with a Pattern Recognition System. Am Lehrstuhl für Mustererkennung bieten wir Praktika an, die mit unserer aktuellen Forschung in den Bereichen Mustererkennung & maschinelles Lernen, medizinische Bildverarbeitung und große Datenanwendungen verbunden sind. Im Gegensatz zu einem Kurs mit festem Thema werden die Themen individuell festgelegt. Das 10 ECTS-Projekt richtet sich an Studierende der Informatik. Auch kleinere 5 ECTS-Projekte sind möglich. Das Ziel dieses Praktikums ist es, die Studenten mit einem Mustererkennungssystem vertraut zu machen.
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 47635	**Project Deep Learning in Multimedia Forensics** Project: Deep learning in multimedia forensics	**10 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	
5	**Inhalt**	Subtle traces in the processing history of an image or video can provide a clue on the recording device, or whether some editing was applied. Multimedia forensics investigates methods to extract these traces from the data. Recent methods in multimedia forensics use deep learning to better adapt to data from the internet. In this project, participants will gather practical experience with deep learning methods in multimedia forensics. Participants will implement published methods from scratch, and do own performance investigations on selected example inputs. On the first meeting on October 28, groups of two students will be formed, and tasks will be distributed. During the project, there are regular consultation hours for status updates and programming support.
6	**Lernziele und Kompetenzen**	Fachkompetenz Anwenden Participants implement deep learning algoorithms in python Analysieren Participants transfer methods from abstract mathematical descriptions in the scientific literature to actual code implementations Evaluieren (Beurteilen) • Participants compare the empirical performances of different forensic detectors • Participants discuss possible reasons for observed performance differences of different forensic detectors Lern- bzw. Methodenkompetenz Participants independently research common implementation tricks and tweaks for deep learning systems from publicly available sources in the internet Selbstkompetenz Participants work towards project deadlines within the framework of a larger programming and evaluation task. Sozialkompetenz • Participants organize their work in groups of two students
7	**Voraussetzungen für die Teilnahme**	Practical experience in python is required. It helps to have experience in the implementation of deep learning algorithms, but this is not required.
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 270 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93111	**Rechnerarchitekturen für Deep-Learning Anwendungen** Computer architectures for deep-learning applications	**10 ECTS**
2	Lehrveranstaltungen	Praktikum: Rechnerarchitekturen für Deep-Learning Anwendungen (8 SWS)	10 ECTS
3	Lehrende	Philipp Holzinger Prof. Dr.-Ing. Dietmar Fey Simon Pfenning	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Dietmar Fey Dr.-Ing. Marc Reichenbach
5	**Inhalt**	Maschinelles Lernen, im speziellen Deep-Learning Netzwerke haben in den letzten Jahren stark an Bedeutung gewonnen. Anwendungsfelder umfassen beispielsweise für die Klassifikation von Bildern, das Verstehen von menschlicher Sprache oder die Überwachung von Rechentechnik bzgl. IT-Angriffen. Während die Grundlagen von Deep-Learning (Neuronale Netze) schon über einen langen Zeitraum erforscht wurden, ist eine flächendeckende Anwendung von Deep-Learning Netzwerken erst seit jüngster Zeit möglich, da nun entsprechende leistungsfähige Rechnerarchitekturen zur Verfügung stehen um die aufwendigen Berechnungen durchzuführen. Das genannte Praktikum beschäftigt sich mit der Evaluierung verschiedener Rechnerarchitekturen (mit entsprechenden Architektureigenschaften) bzgl. der performanten Auswertung von Deep-Learning Netzwerken. Dabei werden die Architekturen CPU, GPU und FPGA genauer untersucht und bewertet. Für eine schnelle Auswertung von Deep-Learning Netzwerken spielt Parallelrechentechnik eine wichtige Rolle, deswegen werden unter anderem folgende Fragen im Praktikum genauer beantwortet: • Wie kann ich Multi-Core CPUs effizient nutzen? • Kann ich mit der Verwendung der SIMD-Extensions (SSE, AVX) ein Beschleunigung der Auswertegeschwindigkeit erzielen? • Was sind GPUs und wie kann ich die massive Parallelität für die Auswertung von Deep-Learning Netzwerken nutzen? • Welche Rolle spielen Caches bzw. die Speicherhierachie eines Rechensystems für die schnelle Auswertung von Deep-Learning Netzwerken? • Wie können FPGAs genutzt werden um Deep-Learning Netzwerke applikationsspezifisch zu implementieren? Für einen einfachen Einstieg wird das Deep-Learning Framework Tensorflow verwendet. Anhand einfacher Besipiele können o.g. Fragen beantwortet werden. Ein größeres Beispiel, gerechnet auf unseren eigenen Servern mit Hardware von AMD, Intel und Nvidia zeigt wie sich reale Anwendungen effizient mit "Customer-Hardware" umsetzen lassen. Hardware in verschiedenen Leistungsklassen (von wenigen Milliwatt bis mehrere Kilowatt) steht dabei zur Verfügung. Die Experimente werden in unserem Parallelrechenlabor durchgeführt.

		Die Studierenden erhalten eine Einführung in Form von Vorträgen. Hauptbestandteil ist die praktische Arbeit mit genannten Rechnerarchitekturen durch effiziente Programmierung mittels C(++), Cuda, OpenCL. Für den erfolgreichen Abschluss des Praktikums ist Kolloquium (ca. 15 Minuten) zu bestehen und ein Bericht (ca. 10 Seiten) anzufertigen.
6	**Lernziele und Kompetenzen**	Verstehen Die Studierenden verstehen die elementaren Grundfunktion neuronaler Netze und wie sich diese auf Hardware abbilden lassen. Die Studierenden lernen die Architekturen CPU, GPU und FPGA kennen, verstehen den internen Aufbau und können die wesentlichen Architektureigenschaften wiedergeben. Analysieren Die Studierenden nutzen CPUs, GPUs und FPGAs für die Auswertung von DL-Netzwerken. Dabei können sie Performancemetriken (Laufzeit, Energieverbrauch) anwenden. Evaluieren Die Studierenden können die Architekturen CPU, GPU und FPGA hinsichtlich ihrer Eignung für die Auswertung von DL-Netzwerken vergleichen.
7	**Voraussetzungen für die Teilnahme**	• Grundlegende Programmierkenntnisse in C • Freude an der Hardware • Detaillierte Kenntnisse in FPGAs, VHDL oder Assembler sind nicht erforderlich • Wissen im Bereich Deep-Learning / Tensorflow ist nicht erforderlich
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 100 h Eigenstudium: 200 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques for Building Intelligent Systems; Aurélien Géron

1	**Modulbezeichnung** 344655	**Secure Web Development**	**10 ECTS**
2	Lehrveranstaltungen	Praktikum/Projekt: Projekt/Praktikum: Secure Web Development (10 SWS)	10 ECTS
3	Lehrende	Philipp Klein	

4	**Modulverantwortliche/r**	
5	**Inhalt**	In diesem Projekt/Praktikum erlernt ihr anhand eines Vorlesungsteils und eines anschliessenden Team-Projekts, Web-Anwendungen zu Erstellen (Backend und Frontend, mit Fokus auf Backend) und auszuliefern (Deployment, DevOps). Inhalte sind u.a.: • Backend-Programmierung mit Python und Django • Frontend-Programmierung mit CSS (vermutl. Bootstrap) und Javascript • Continuous Integration und Continous Delivery mit GitlabCI und Ansible • Aufsetzen eines Staging- und Production-Servers • Sichere Entwicklung im Open-Source-Umfeld • Angriffe auf Webapplikationen und "Abwehr" dieser In diesem Praktikum/Projekt agieren wir wie ein Software-Unternehmen. Es wird einen Auftraggeber, ein Lasten- und Pflichtenheft sowie Meilensteine und Deadlines geben. Die Arbeit geschieht in einem Team mit einem woechentlichen Meeting. Ort und Zeit kann das Team selbst bestimmen. In der initialen Lernphase bekommt ihr Vorlesungsmaterial und Uebungen zur Verfuegung gestellt. Das Videomaterial umfasst sowohl Frontend und -Backend-Programmierung als auch DevOps. In der anschliessenden Projektphase ist eine Spezialisierung auf einen Bereich moeglich.
6	**Lernziele und Kompetenzen**	Fachkompetenz Erschaffen Studierende planen und entwickeln Webapplikationen in Python und Django. Sie legen in der Konzeption und Umsetzung grossen Wert auf Sicherheit. Lern- bzw. Methodenkompetenz Studierende wenden sämtliche Arbeitsschritte in der Entwicklung von Webapplikationen, von der Planung über die Erstellung einer CI-Pipeline bis zur Auslieferung, an. Selbstkompetenz Studierende integrieren sich in Teams, planen das gemeinsame Vorgehen und reflektieren über das bisher Geleistete. Sozialkompetenz Studierende dokumentieren ihre Arbeit verständlich und nachvollziehbar. Sie kommunizieren innerhalb des Teams respektvoll und wertschätzend.
7	**Voraussetzungen für die Teilnahme**	Grundkenntnisse in Python
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!

9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio Die Note berechnet sich aus der geleisteten Arbeit im Team und dem Abschlussvortrag. Die Gewichtung liegt bei 90:10. Bei der Teamarbeit wird sowohl der Code, als auch das Verhalten im Team bewertet.
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93162	**Softwareentwicklungspraktikum Lehramt** Software development lab for teaching degree students	**10 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Prof. Dr. Marc-Pascal Berges
5	**Inhalt**	Agile Entwicklung eines Projektes/Programms für den Einsatz im Unterricht Organisation eines größeren Projektes Zeitmanagement Kommunikation in der Gruppe
6	**Lernziele und Kompetenzen**	Die Studierenden - entwickeln in Gruppen ein lauffähiges Programm für den Einsatz im Unterricht - verwalten ihren Projektfortschritt mit Hilfe eines Projektboards - lernen agile Entwicklung und deren Einsatzmöglichkeit im Unterricht kennen
7	**Voraussetzungen für die Teilnahme**	Es wird der erfolgreiche Abschluss der folgenden Module als Voraussetzung empfohlen: • Einführung in die Programmierung • Einführung in die Algorithmik • Einführung in Datenbanken • Einführung in das Software Engineering
8	**Einpassung in Studienverlaufsplan**	Semester: 7;5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung Prüfungsleistung: Entwicklung und Vorstellung eines Softwareprojektes in Gruppen. Die Vorstellung erfolgt in einem 20-30minütigem Vortrag. Die Entwicklung wird durch einen 30-60 seitigen Praktikumsbericht dokumentiert.
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 120 h Eigenstudium: 180 h
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Best Practice Software-Engineering, A.Schatten, Spektrum Verlag

		Software-Qualität, Dirk W. Hoffmann, Springer Verlag

1	**Modulbezeichnung** 182798	**Supercomputing Praktikum** Student cluster competition	**10 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Dietmar Fey
5	**Inhalt**	Das Supercomputer Praktikum bereitet die Studierenden auf eine Teilnahme an der Student Cluster Competition(SCC). Im Rahmen des Wettbewerbs muss ein Teil eines Supercomputers so eingerichtet werden, dass klassische Benchmarks und andere HPC-Anwendungen möglichst performant und effizient ausgeführt werden können. Die Studierenden bearbeiten dafür im Semester die folgenden Themen: - Aufbau und Komponenten von Supercomputern - Benchmarking von Anwendung bezüglich Rechenleistung und Energieverbrauch in HPC-Systemen - Konfiguration, Administration und Debugging von Supercomputern Des Weiteren lernen die Studierenden sich selbstständig als Team zu organisieren und den Ablauf und Teilnahme eines Bewerbungsprozesses in einem professionellen Umfeld.
6	**Lernziele und Kompetenzen**	Studierende... - haben alle notwendigen Fähigkeiten erworben, um an dem Student Cluster Competition auf der SC oder ISC teilzunehmen, - können einen Cluster-Computer ingenieurmäßig entwerfen und zusammenbauen, - verstehen und konfigurieren die speziellen Hardware-Architekturen von HPC-Systemen, - können HPC-Anwendungen installieren, messen und optimieren, - können den Zustand des Systems zu überwachen und es bei Bedarf zu reparieren.
7	**Voraussetzungen für die Teilnahme**	- Grundkenntnisse der UNIX-Systeme und der Programmierung in C oder C++ - Bereitschaft in die USA zu reisen um am Wettbewerb teilzunehmen (Reisekosten werden übernommen)
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	nur im Sommersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 100 h Eigenstudium: 200 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	

Hauptseminar

1	**Modulbezeichnung** 93655	**Advanced Competitive Programming** Advanced competitive programming	**5 ECTS**
2	Lehrveranstaltungen	Seminar: Advanced Competitive Programming (2 SWS)	5 ECTS
3	Lehrende	Paul Wild	

4	**Modulverantwortliche/r**	Paul Wild
5	**Inhalt**	Es werden verschiedene Algorithmen, algorithmische Methoden und Datenstrukturen vorgestellt, wie sie insbesondere im Kontext von Programmierwettbewerben wie dem ICPC zur Anwendung kommen. • Algorithmen zur schnellen Polynommultiplikation • Siebmethoden zur Berechnung zahlentheoretischer Funktionen • Datenstrukturen zur Verarbeitung intervallbasierter Anfragen auf Arrays und Bäumen • Methoden zur Optimierung Dynamischer Programmierung • Sweep-Verfahren aus der Algorithmischen Geometrie • Suffixarrays und Anwendungen Bei etwa der Hälfte der Termine finden jeweils Vorträge statt, in denen die verschiedenen Konzepte und Algorithmen durch die Studierenden vorgestellt werden. Bei den verbleibenden Terminen werden diese von den Studierenden umgesetzt bzw. implementiert und in Übungsaufgaben zur Anwendung gebracht. Dabei demonstrieren und erklären die Studierenden die Arbeitsweise ihrer Implementierung.

6	**Lernziele und Kompetenzen**	•
		  ° Wissen: Die Studierenden geben Probleme aus dem Bereich der Wettbewerbsprogrammierung und die zu ihrer Lösung verwendeten Algorithmen wieder.
		  ° Analysieren: Die Studierenden unterscheiden verschiedene Algorithmen hinsichtlich ihrer Funktionsweise und Laufzeitkomplexität und begründen deren Korrektheit.
		  ° Anwenden: DIe Studierenden implementieren die gelernten Algorithmen eigenständig und sauber, sodass diese die jeweiligen Probleme korrekt und innerhalb vorgegebener Zeitschranken lösen.
		•
		  ° Die Studierenden erarbeiten sich die Konzepte eigenständig aus den zur Verfügung gestellten Materialien und geben diese in einem strukturierten Vortrag klar und verständlich wieder.
		•
		  ° Die Studierenden arbeiten in einer größeren Gruppe an komplexen Problemen und erläutern eigene Einsichten den Mitstudierenden in verständlicher Form.

7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4

9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• A. Laaksonen: Guide to Competitive Programming, Springer, 2017. • F. Halim und S. Halim: Competitive Programming 3, the new lower bound of programming contests, Lulu.com, 2013. • T. Cormen et al.: Introduction to Algorithms, MIT Press, 2001. • J. Erickson: Algorithms, self published, 2019.

1	**Modulbezeichnung** 93126	**Automaten über unendlichen Wörtern** Automatons over infinite words	**5 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt! Es besteht keine Anwesenheitspflicht; regelmäßige aktive Teilnahme wird aber erwartet.	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Prof. Dr. Lutz Schröder
5	**Inhalt**	• Büchi-, Street- und Muller-Automaten • Unendliche Spiele und Determiniertheit • Determinisierung, Safra/Piterman-Konstruktion • Komplementierung • Alternierung • Baumautomaten • Modaler mu-Kalkül • Monadic Second Order Logic
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben grundlegende Definitionen und Resultate der Theorie der unendlichen Spiele und Automaten über unendlichen Strukturen wieder. Verstehen Die Studierenden erläutern grundlegende algorithmische Verfahren für Automaten und Spiele. Anwenden Die Studierenden teilen Automaten- und Spielbegriffe hinsichtlich ihrer Komplexität und ihrer Ausdrucksstärke ein. Erschaffen Die Studierenden bereiten Resultate des Gebiets und ihre Herleitung eigenständig zur Präsentation auf. Lern- bzw. Methodenkompetenz Die Studierenden erarbeiten sich selbständig den Inhalt wissenschaftlicher Veröffentlichungen und geben diesen in einem strukturierten Vortrag verständlich wieder. Sozialkompetenz Die Studierenden arbeiten in einer größeren Gruppe an komplexen Problemen und erläutern eigene Einsichten den Mitstudierenden in verständlicher Form.
7	**Voraussetzungen für die Teilnahme**	Das Seminar setzt Kenntnisse in Automatentheorie voraus.
8	**Einpassung in Studienverlaufsplan**	Semester: 4;5;6
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung Die Prüfungsleistung besteht in einer Ausarbeitung und einem 90-minütigen erfolgreichen Vortrag.
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h

14	**Dauer des Moduls**	1 Semester Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Erich Grädel, Wolfgang Thomas, Thomas Wilke: Automata, Logics, and Infinite Games: A Guide to Current Research. Lecture Notes in Computer Science 2500, Springer, 2002. Nir Piterman: From Nondeterministic Büchi and Streett Automata to Deterministic Parity Automata. Logical Methods in Computer Science 3(3) (2007) Oliver Friedmann, Martin Lange: The Modal mu-Calculus Caught Off Guard. TABLEAUX 2011: 149-163 Damian Niwinski, Igor Walukiewicz: Games for the mu-Calculus. Theor. Comput. Sci. 163: 99-116 (1996) _____

1	**Modulbezeichnung** 921878	**Blender Seminar** Blender seminar	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Marc Stamminger
5	**Inhalt**	In diesem Modul vermitteln wir grundlegende Kenntnisse über den Umgang mit 3D-Modellierungstools und zur Planung von Projekten anhand der OpenSource Software BlendER.
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Studierende können... • ... mathematische Grundlagen verschiedener Beleuchtungsmodelle darlegen • ... Konzepte der Szenengestaltung und Beleuchtung reproduzieren • ... Mathematische Grundlagen zu Interpolationsverfahren darlegen • ... sich an Lizenzmodellen für eigene Werke erinnern • ... über verschiedene Strategien zur Projekt- und Teamplanung berichten Verstehen Studierende können... • ... verschiedene Verfahren zur Keyframe- oder Vertexinterpolation schildern • ... Probleme bei der Erstellung von Geometrie aufzeigen • ... Probleme des Photon-Tracing illustrieren Anwenden Studierende können... • ... Wissen über Szenengestaltung und Beleuchtung auf konkrete Beispiele anwenden • ... eigene Materialshader auf der Grundlage vorgegebener Formeln/Vorschriften in BlendER umsetzen Analysieren • Selbstständige Projektplanung, Teamorganisation und Umsetzung eines 3D-Projektes in vorgegebenem Zeitfenstern
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung

11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 80 h Eigenstudium: 70 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 47637	**Geschichte der Rechentechnik** History of computing	**5 ECTS**
2	Lehrveranstaltungen	Hauptseminar: Die Geschichte der Rechentechnik I (2 SWS) Übung: Übung zu Geschichte der Rechentechnik (2 SWS)	5 ECTS -
3	Lehrende	Dr.phil. Felix Schmutterer	

4	**Modulverantwortliche/r**	Dr.phil. Felix Schmutterer
5	**Inhalt**	Gegenstand des Seminars sind die Meilensteine der Rechentechnik" ausgehend vom 19. Jahrhundert. Diese einschneidenden Entwicklungen von Rechenmaschinen zu ersten Werkzeugen der Datenverarbeitung werden zunächst den Ausgangspunkt bilden. Turingmaschinen" und die neuen Bedürfnisse" von Militär wie etwa Chiffrierung und De-Chiffrierung werden dann zentrale Themen des Seminars bilden. Im Fokus steht dabei stets die Funktionsweise der Maschinen. Darüber hinaus werden die Rechner konsequent im Kontext ihrer Zeit diskutiert werden. Insbesondere wird dabei auf die steigenden Anforderungen und die veränderlichen Einsatzmöglichkeiten wie etwa im Falle der Enigma einzugehen sein. Die genauen Themen werden zu Semesterbeginn festgelegt. Die Themenliste kann beim Dozenten erfragt werden.
6	**Lernziele und Kompetenzen**	Die Studierenden • recherchieren und arbeiten mit historischen Quellen und wissenschaftlicher Literatur aus den Bereichen Informatik und Geschichte • beschreiben Aspekte der Rechentechnik • erarbeiten sich die Fähigkeit, wichtige Aspekte für einen wissenschaftlichen Vortrag darzustellen und strukturieren diesen • vertreten ihre Auffassung in einer Diskussion und hinterfragen ihr Thema • konzipieren und formulieren eine schriftliche Zusammenfassung des Vortrags
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung (30 Minuten)
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	in jedem Semester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 47614	**Green AI - AI for sustainability and sustainability of AI**	**5 ECTS**
2	Lehrveranstaltungen	Hauptseminar: Green AI - AI for Sustainability and Sustainability of AI (2 SWS)	5 ECTS
3	Lehrende	Prof. Dr. Björn Eskofier Eva Dorschky René Raab	

4	**Modulverantwortliche/r**	Eva Dorschky Prof. Dr. Björn Eskofier
5	**Inhalt**	Can we use AI to combat global climate change? How can advances in machine learning and data science help to monitor climate crises and to conserve nature? What is the role of AI in reducing greenhouse gas emissions in the manufacturing industries, transportation infrastructure, agriculture, and power sector? In this seminar, we will develop and discuss future perspectives of AI for sustainability, considering the sustainability of AI itself. Current advances in machine learning, particularly deep learning, are enabling new applications but are accompanied by an exponential increase in computational cost and thus significant carbon emissions (Schwartz et al., 2020; Vinuesa et al., 2020). In this seminar, we will learn about important aspects of improving the sustainability of machine learning algorithms. This seminar offers a different perspective on machine learning as taught in other courses, namely its role in global climate change. This aspect is becoming increasingly important in research, but also in industry. Therefore, this seminar provides the following items: • Introduction to "Green AI" versus "Red AI" • Guests talks on related research topics • Group discussions on future prospects of AI, specifically machine learning • Best practices for literature review and scientific presentations • Literature review on Green AI in certain areas in groups • Scientific talk of each student on one specific topic
6	**Lernziele und Kompetenzen**	Students will analyze • the opportunities that AI offers to combat global climate change • the negative impact of AI on global climate change • current research topics in the field of "Green AI" Students will be able to • discuss and work in a group • perform and write a literature review • give a scientific presentation
7	**Voraussetzungen für die Teilnahme**	Basic knowledge in machine learning is required to take part in the seminar. Students are expected to have completed one or more basic courses, such as PR, PA, IntroPR, DL, MTLS, or equivalent.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!

9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Schwartz, Roy et al. (2020). "Green ai. In: Communications of the ACM 63.12, pp. 54 63. Vinuesa, Ricardo et al. (2020). "The role of artificial intelligence in achieving the Sustainable Development Goals. In: Nature communications 11.1, pp. 110.

1	**Modulbezeichnung** 396551	**IT-Sicherheits-Seminar** IT security seminar	**5 ECTS**
2	Lehrveranstaltungen	Seminar: IT-Sicherheits-Konferenzseminar (Bachelor) (0 SWS)	5 ECTS
3	Lehrende	Christian Eichenmüller Prof. Dr.-Ing. Felix Freiling	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Felix Freiling
5	**Inhalt**	Wechselnde Themen aus dem Bereich IT-Sicherheit.
6	**Lernziele und Kompetenzen**	Teilnehmende arbeiten sich selbständig anhand wissenschaftlicher Literatur in ein vorgeschlagenes oder nach Absprache frei gewähltes Thema aus dem Bereich der IT-Sicherheit ein, erstellen dazu eine schriftliche Ausarbeitung, bereiten einen Seminarvortrag vor und halten ihn. Zu den hier zu erwerbenden Kompetenzen zählen die Literaturrecherche, korrektes Zitieren, die Fähigkeit auszuwählen, welche Aspekte in der Ausarbeitung und im Vortrag behandelt werden, zielgruppengerechtes Schreiben sowie der Umgang mit dem Textsatzsystem LaTeX. In jedem Wintersemester findet das Seminar als Konferenzseminar statt. Hier üben die Studierenden dann auch das Prozedere ein, das beim Einreichen einer wissenschaftlichen Arbeit bei einer Konferenz üblich ist: Unter anderem lernen sie, die Arbeiten anderer Personen im Review-Prozess zu beurteilen und Kritik und Verbesserungsempfehlungen auszusprechen sowie für die eigene Arbeit anzunehmen und umzusetzen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 44590	**Konzepte von Betriebssystem-Komponenten**	**5 ECTS**
2	Lehrveranstaltungen	Hauptseminar: Seminar Ausgewählte Kapitel der Systemsoftwaretechnik (SWS)	-
3	Lehrende	Peter Wägemann Eva Dengler	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93096	**Iterative Lösungsverfahren für lineare und nichtlineare Gleichungssysteme** Iterative solution methods for linear and non-linear systems	**5 ECTS**
2	Lehrveranstaltungen	Hauptseminar: Mathematisches Seminar "Nichtlineare Löser" (2 SWS)	5 ECTS
3	Lehrende		

4	**Modulverantwortliche/r**	Prof. Dr. Christoph Pflaum
5	**Inhalt**	• Einführung in die einzelnen Phasen der Softwareentwicklung: Anforderungsanalyse, Spezifikation, Entwurf, Implementierung, Test, Wartung • Beispielhafter Einsatz ausgewählter repräsentativer Verfahren zur Unterstützung dieser Entwicklungsphasen • Ergonomische Prinzipien Benutzungsoberfläche • Objektorientierte Analyse und Design mittels UML • Entwurfsmuster als konstruktive, wiederverwendbare Lösungsansätze für ganze Problemklassen • Automatisch unterstützte Implementierung aus UML-Diagrammen • Teststrategien • Refactoring zur Unterstützung der Wartungsphase
6	**Lernziele und Kompetenzen**	Die Studierenden • wenden auf Basis der bereits erworbenen Programmierkenntnisse systematische und strukturierte Vorgehensweisen (wie das Wasserfall- und V-Modell) zur Bewältigung der Komplexität im Zusammenhang mit dem Programmieren-im-Großen" an; • benutzen ausgewählte Spezifikationssprachen (wie Endliche Automaten, Petri-Netze und OCL), um komplexe Problemstellungen eindeutig zu formulieren und durch ausgewählte Entwurfsverfahren umzusetzen; • wenden UML-Diagramme (wie Use Case-, Klassen-, Sequenz- und Kommunikationsdiagramme) zum Zweck objektorientierter Analyse- und Design-Aktivitäten an; • reproduzieren allgemeine Entwurfslösungen wiederkehrender Probleme des Software Engineering durch Verwendung von Entwurfsmustern; • erfassen funktionale und strukturelle Testansätze; • setzen Refactoring-Strategien zur gezielten Erhöhung der Software-Änderungsfreundlichkeit um.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 3
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	
16	**Literaturhinweise**	Lehrbuch der Softwaretechnik (Band 1), Helmut Balzert, 2000

1	**Modulbezeichnung** 93127	**Maschinelles Lernen: Einführung** Machine learning: Introduction	**5 ECTS**
2	Lehrveranstaltungen	Hauptseminar: Machine Learning: Introduction (2 SWS)	5 ECTS
3	Lehrende	Prof. Dr. Michael Philippsen Tobias Feigl	

4	**Modulverantwortliche/r**	Prof. Dr. Michael Philippsen
5	**Inhalt**	Dieses Seminar führt in das Themengebiet des maschinellen Lernens (ML) ein. ML ist die Wissenschaft, Computer zum Handeln zu bewegen, ohne explizit programmiert zu werden. ML ist heute so allgegenwärtig, dass wir es wahrscheinlich täglich verwenden, ohne es zu wissen. So hat ML in den letzten Jahren beispielsweise selbstfahrende Autos, praktische Bild- und Spracherkennung und die effektive Partner- und Websuche ermöglicht. Ziel des Seminars ist eine umfassende Einführung in das maschinelle Lernen, Analyse und Verarbeitung von Daten sowie statistische Mustererkennung. Zu den Themen gehören: (1) Klassifizierungs- und Regressionsprobleme; (2) überwachtes Lernen (parametrische und nicht parametrische Algorithmen, lineare und logistische Regression, k-nächster Nachbar, Support-Vector-Machines, Entscheidungsbäume, flache neuronale Netze); (3) unüberwachtes Lernen (K-Means, Clustering, Dimensionsreduktion, PCA, LDA, Empfehlungssysteme); (4) Ensemble- und Online-Lernen; (5) Regularisierung: Modelldiagnose, Fehleranalyse und Qualitätsmetriken sowie Interpretation der Ergebnisse; (5) evolutionäre Algorithmen; (6) Anomalieerkennung und Gaußsche Verteilungen; (7) Bayes, Kalman-Filter und Gaußsche Prozesse. Die genannten Themen sind an den aktuellen Forschungsstand angepasst und wechseln sich jährlich ab. Das Seminar gibt einen Einblick in die Welt des maschinellen Lernens und befähigt den Studierenden eine wissenschaftliche Präsentation und Ausarbeitung anzufertigen, um individuell erworbenes Wissen einem Fachpublikum vermitteln zu können.
6	**Lernziele und Kompetenzen**	Die Teilnahme an diesem Seminar ermöglicht den Studierenden sich in der Kompetenz maschinelles Lernen auszubilden und erlerntes Wissen in Form einer angeleiteten Präsentation und Ausarbeitung wissenschaftlich darzustellen und zu kommunizieren: Die Studierenden erlangen durch das Seminar die Kompetenz und das Wissen: • prinzipielle Vorgehensweisen beim maschinellen Lernen zu erläutern, • Vor- und Nachteile einzelner Methoden zu untersuchen, • Chancen und Grenzen des maschinellen Lernens zu erläutern, • Sachverhalte unter Fachleuten zu diskutieren, • fachspezifische Fragen für das Gebiet zu beantworten, • Konzepte des Maschinellen Lernens im Allgemeinen und deren Anwendung in Applikationsgebieten der Industrie, Sozialwesen, Bildung und Sport zu erlernen,

		• Datenvorverarbeitung, ML-Methoden und Interpretation der Ergebnisse in konkreten Fragestellungen zu modellieren und zu adaptieren. Weiter führt das Seminar die Studierende in das wissenschaftliche Arbeiten ein, um selbstständig: • erforderliche Literatur aufzufinden, zu analysieren und zu bewerten, • sich eigenständig in ein Themengebiet einzuarbeiten, • Grundzüge der Präsentationstechniken anzuwenden und zu motivieren, • eine Präsentation mit Begleitmaterial für ein Fachpublikum zu entwickeln, • einen Vortrag passend für einen vorgegebenen Zeitrahmen durchzuführen, • eine Ausarbeitung im Stil einer wissenschaftlichen Publikation mit Latex anzufertigen, • Sprache, Sprachangemessenheit, Inhalt sowie Aufbau und die wissenschaftliche Darstellung einer wissenschaftlichen Ausarbeitung zu verinnerlichen, • und die eigene Kognition und Kreativität in der Ausarbeitung zu bewerten.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• A. Müller und S. Guido: Introduction to Machine Learning with Python: A Guide for Data Scientists, O'Reilly UK Ltd., 2016 • K. P. Murphy: Machine learning - a probabilistic perspective, Adaptive computation and machine learning series, MIT Press, 2012. • T. J. Hastie und R. Tibshirani und J. H. Friedman: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics, 2009.

		• T. M. Mitchell: Machine Learning, McGraw-Hill Education Ltd., 1997 • F. V. Jensen: An Introduction To Bayesian Networks, Springer, 1996 • J. A. Freeman: Simulating neural networks - with Mathematica, Addison-Wesley Professional, 1993 • J. A. Hertz und A. Krogh und R. G. Palmer: Introduction to the theory of neural computation, Westview Press, 1991 • R. Rojas: Theorie der neuronalen Netze - eine systematische Einführung, Springer, 1993 • W. Banzhaf und F. D. Francone und R. E. Keller und P. Nordin: Genetic programming - An Introduction: On the Automatic Evolution of Computer Programs and Its Applications, Morgan Kaufmann, 1998 • M. Mitchell: An introduction to genetic algorithms, MIT Press, 1996 • Z. Michalewicz: Genetic Algorithms + Data Structures = Evolution Programs, Springer, 1992 • M. Bishop: Pattern Recognition and Machine Learning (Information Science and Statistics), Springer, 2006

1	**Modulbezeichnung** 304439	**Nailing your Thesis (VUE 5-ECTS)** Nailing your thesis (VUE 5-ECTS)	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Dirk Riehle
5	**Inhalt**	This course teaches students how to perform scientific research for their final thesis or a research paper. The goal is to prepare students for a Bachelor or Master research thesis. The course covers the following topics: • Science and society • The research process • Theory building research • Theory validation research • Writing a research thesis/paper • The scientific community Students can choose one or both of two components: • VUE (VL + UE or seminar), 4 SWS, 5 ECTS. VUE combines lectures with homework and exercises. VUE is run as a 3h block. • PROJ (small research project), 2 SWS, 5 ECTS. In PROJ, students perform a small research project, either individually or in teams. The available projects will be presented at the beginning of the course. Students perform the research, write a paper, and hold a presentation about their work. Sign-up and further course information are available at https://nyt.uni1.de - please sign up for the course on StudOn (available through previous link) as soon as possible. The course information will also tell you how the course will be held (online or in person).
6	**Lernziele und Kompetenzen**	• Students gain an understanding of how science works • Students learn how to perform research work • Students learn how to write a research thesis
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Sommersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	See https://nyt.uni1.de

1	**Modulbezeichnung** 941318	**Neuartige Rechnerarchitekturen**	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Dietmar Fey
5	**Inhalt**	Die Entwicklung moderner CPUs hat eine interessante Evolution durchlaufen. Angefangen bei einfachen Single-Core CPUs wurde zunächst die Taktschraube immer weiter nach oben gedreht. Als dies aus thermischen Grund nicht weiter möglich war, wurden Parallelrechner aus ihrer akademischen Nische vertrieben und zum Allgemeingut eines jeden Informatikers. Neuere Entwicklung zeigen nun den Einsatz von heterogenen Rechnerarchitekturen, also die Verbindung verschiedener Recheneinheiten wie CPUs, GPUs, FPGAs, um mittels Spezialhardware anfallende Aufgaben schneller und energieeffizienter lösen zu können. Neuste Forschungsansätze hingegen versuchen nun auch den Hauptspeicher eines Rechners "intelligent" zu machen und Prozessoren direkt in den Speicher zu integrieren - sogenanntes in- oder near-memory-Computing. Ziel dieses Moduls ist das ... ... kennen, ... verstehen, ... verwenden, ... vergleichen, und evaluieren verschiedener Rechnerarchitekturen von der Multi-Core CPU bis zum FPGA-Near-Memory-Beschleuniger. Anhand praktischer Anwendungen (z.B. Neuronale Netze, Bildverarbeitung, Autonomes Fahren) können die Architekturen erprobt werden. Hierzu wird jedem Teilnehmenden ein Thema/Architektur zur Bearbeitung übertragen, welche sie/er selbstständig wissenschaftlich in einer schriftlichen Ausarbeitung und didaktisch in einem Vortrag aufarbeitet und präsentiert.
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Lernende können Wissen über die Grundprinzipien moderner Rechnerarchitekturen (Intel, ARM CPUs; AMD, Nvidia GPUs; FPGAs, Beschleunigerkerne) wiedergeben. Verstehen Lernende verstehen die Grundprinzipien der Datenverarbeitung der einzelnen Architekturen; im speziell verstehen sie ob und warum eine vorgegebene Architektur besonders gut für die Lösung eines Problems geeignet ist. Lernende verstehen die unterschiedlichen Ansätze zur Parallelismus der vorgestellten Architekturen. Anwenden

		Lernende sind in der Lage Anwendungen auf den vorgegebenen Architekturen z.B. durch Programmierung umzusetzen. Hierzu erklären Studierende wie die Parallelisierungstechniken in bestehenden Architekturen eingesetzt werden. Evaluieren (Beurteilen) Lernende evaluieren die Eignung von Architekturen, bestimmte Probleme effizient auf diese Abbilden zu können. Sozialkompetenz Lernende können komplexe fachbezogene Inhalte klar und zielgruppengerecht präsentieren und eigene Standpunkte in einer Fachdiskussion argumentativ vertreten.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93144	**The AMOS Project (PO Role, SEMI 5 ECTS)**	**5 ECTS**
2	Lehrveranstaltungen	Übung: The AMOS Project (UE) (2 SWS) Vorlesung: The AMOS Project (VL) (2 SWS)	- -
3	Lehrende	Prof. Dr. Dirk Riehle	

4	**Modulverantwortliche/r**	Prof. Dr. Dirk Riehle
5	**Inhalt**	This course teaches agile methods (Scrum and XP) and open source tools using a single semester-long project. Topics covered are: • Agile methods and related software development processes • Scrum roles, process practices, including product and engineering management • Technical practices like refactoring, continuous integration, and test-driven development • Principles and best practices of open source software development The project is a software development project in which each student team works with an industry partner who provides the idea for the project. This is a practical hands-on experience. Students can play one of two primary roles: • Product owner. In this function, a student defines, prioritizes, communicates, and reviews requirements. The total effort adds up to 5 ECTS. • Software developer. In this function, a student estimates their effort for requirements and implements them. The total effort adds up to 10 ECTS. Students will be organized into teams of 7-8 people, combining product owners with software developers. An industry partner will provide requirements to be worked out in detail by the product owners and to be realized by the software developers. The available projects will be presented in the run-up to the course. Class consists of a 90min lecture followed by a 90min team meeting. Rooms and times for team meetings are assigned at the beginning of the semester. You must be able to regularly participate in the team meetings. If you can't, do not sign up for this course. Students choosing the software developer role must have prior software development experience. Sign-up and further course information are available at https://amos.uni1.de - please sign up for the course on StudOn (available through previous link) as soon as possible. The course information will also tell you how the course will be held (online or in person).
6	**Lernziele und Kompetenzen**	• Students learn about software products and software development in an industry context • Students learn about agile methods, in particular, Scrum and Extreme Programming

		• Students gain practical hands-on experience with a Scrum process and XP technical practices
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 404439	**Nailing your Thesis (PROJ 5-ECTS)** Nailing your thesis (PROJ 5-ECTS)	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Dirk Riehle
5	**Inhalt**	This course teaches students how to perform scientific research for their final thesis or a research paper. The goal is to prepare students for a Bachelor or Master research thesis. The course covers the following topics: • Science and society • The research process • Theory building research • Theory validation research • Writing a research thesis/paper • The scientific community Students can choose one or both of two components: • VUE (VL + UE or seminar), 4 SWS, 5 ECTS. VUE combines lectures with homework and exercises. VUE is run as a 3h block. • PROJ (small research project), 2 SWS, 5 ECTS. In PROJ, students perform a small research project, either individually or in teams. The available projects will be presented at the beginning of the course. Students perform the research, write a paper, and hold a presentation about their work. Sign-up and further course information are available at https://nyt.uni1.de - please sign up for the course on StudOn (available through previous link) as soon as possible. The course information will also tell you how the course will be held (online or in person).
6	**Lernziele und Kompetenzen**	• Students gain an understanding of how science works • Students learn how to perform research work • Students learn how to write a research thesis
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)

12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 150 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	See https://nyt.uni1.de

1	**Modulbezeichnung** 93186	**Seminar Applied Software Engineering** Seminar: Applied software engineering	**5 ECTS**
2	Lehrveranstaltungen	Seminar: Seminar Applied Software Engineering (0 SWS) Hauptseminar: Applied Software Engineering Master-Seminar (SWS)	5 ECTS -
3	Lehrende		

4	**Modulverantwortliche/r**	Prof. Dr. Dirk Riehle
5	**Inhalt**	This module lets students fulfill their degree program's seminar obligation by fulfilling a seminar topic in software engineering and/or open source. We prefer that you use one of our existing courses for your seminar obligation, but are willing to have you for a one-off topic if none of our courses fit. Seminar topics should be in the domain of (applied) software engineering and may or may not include open source software as a topic. You can find current seminar / project / thesis topics at https://oss.cs.fau.de/fun ; all topics are customizable to your needs (ECTS points). If you find something that interests you, please talk to the respective person listed in the topic description (bottom of document, usually).
6	**Lernziele und Kompetenzen**	• Students learn to analyze and summarize a software engineering topic and to present it in class
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 150 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93179	**Recent Advances in Cryptography** Recent advances in cryptography	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Dominique Schröder
5	**Inhalt**	In this seminar we are exploring recent advances in the area of cryptography. The students are required to familiarize themselves with recent scientific publications, give a 45 minute talk, and provide a summary of at least 4 pages counted without pictures. After each talk there will be a discuccsion about the presented paper.
6	**Lernziele und Kompetenzen**	On successfully passing the course, the students understand how to work with scientific literature and to present the topics in an understandable way. Furthermore, they learn about measures of privacy and the application thereof with privacy enhancing technologies.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 20 h Eigenstudium: 130 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Selected publications

1	**Modulbezeichnung** 93188	**Research Data Skills** Research data skills	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Dr. Marcus Walther
5	**Inhalt**	Einstieg in Elemente des Forschungsdatenmanagements Der nachhaltige Umgang mit Forschungsdaten gehört heute zum Handwerkszeug und nimmt bei Forschungsprojekten einen immer größeren Raum ein. In diesen Seminar werden folgende Themen bearbeitet: • FAIR-Prinzipien • Versionsverwaltung mit GIT und gitlab • Forschungsinformationen • Datenablage, -dokumentation und -übergabe • scientific workflows • Nutzung von Repositorien • Urheberrecht an Daten und Software • ... offene Themen im Bereich FDM ... Die Studierenden erstellen zum gewählen Thema Präsentationen im Umfang von 15-30 Minuten.
6	**Lernziele und Kompetenzen**	• Einarbeitung in die Fragestellung • Zusammenarbeit mit beteiligten Wissenschaftler/innen • Verständnis komplexer Workflows im Lebenszyklus digitaler Forschungsdaten und Evaluierung der praktischen Machbarkeit • Identifikation von bestehenden Hindernissen und Entwicklung von Lösungsansätzen (organisatorisch, Entwicklung von Software-Tools in Prototypenstadium, ...) • Dokumentation und Präsentation der ermittelten Ergebnisse
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Die Literatur wird je nach Thema vergeben.

1	**Modulbezeichnung** 93204	**Cryptography in Secure Messaging: Understanding and Enhancing Signal** SecMsg	**5 ECTS**
2	Lehrveranstaltungen	Seminar: Cryptography in Secure Messaging: Understanding and Enhancing Signal (4 SWS, WiSe 2023)	5 ECTS
3	Lehrende	Prof. Dr. Paul Rösler	

4	**Modulverantwortliche/r**	Carina Köhner Prof. Dr. Dominique Schröder
5	**Inhalt**	The purpose of this seminar is to obtain an overview of modern messaging protocols used in applications like WhatsApp and Signal. For this, we start with scientific publications that analyze Signal's Double-Ratchet protocol, continue with publications that propose enhancements for it, and end with publications that investigate the strongest possible security messaging protocols can theoretically provide. Each participant will present the core idea of one publication. Furthermore, each student embeds their publication into the related work by explaining the direct relation to an earlier publication. The presentation will consist of a 45 minute talk and a 15 minute discussion. Additionally, each student has to hand in a summary of at least 4 pages counted without pictures. Of these 4 pages at least 1 page explains the relation to a publication presented by another student in this seminar.
6	**Lernziele und Kompetenzen**	On successfully passing the course, the students understand how to work with scientific literature, how to present the core idea of a single publication, and how to explain such an idea in the context of a broader research domain. Furthermore, they obtain an overview of modern key-exchange and messaging protocols used by billions of users every day.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 3
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)

15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Selected publications

1	**Modulbezeichnung** 871263	**Seminar Algorithmische Schönheiten - Algorithms Unplugged** Seminar: Beautiful algorithms - Algorithms unplugged	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Rolf Wanka
5	**Inhalt**	Viele Algorithmen lösen nicht "einfach nur" das Problem, für das sie ausgedacht worden sind, sie sind oft auch ästhetisch sehr ansprechend. Sie benutzen anschauliche Ideen auf überraschend schlaue Art und Weise, oder sie verwenden Ideen aus einem Bereich bewundernswert clever in einem anderen Einsatzbereich wieder. Ziel dieses Seminares ist es, einige dieser Algorithmen kennenzulernen. Die Teilnehmerinnen und Teilnehmer bekommen Texte, lesen diese und suchen zusätzlich einige der Hintergrundaufsätze und stellen "ihre" Algorithmen in einem Vortrag und einer schriftlichen Ausarbeitung vor.
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben wieder, wie sie die neu erlernten Begrifflichkeiten und Verfahren in die bislang im Studium kennengelernten Abläufe und Themenkomplexe einordnen, in dem sie kurz benötigte Fakten darlegen. Verstehen Die Studierenden präsentieren ggf. konkrete Beispiele, auf die die neu erworbenen Kenntnisse angewandt werden, und skizzieren die Vor- und Nachteile der neu erlernten Inhalten. Anwenden Die Studierenden vergleichen die neu erlernten Inhalte mit zuvor, ggf. im gleichen Seminar kennengelernten Inhalten und beschreiben ggf. neue Anwendungsgebiete. Analysieren Die Studierenden bewerten die neu erlernten Inhalte, indem sie wissenschaftlich belastbare Aussagen treffen und mit den Teilnehmerinnen und Teilnehmern diskutieren. Evaluieren (Beurteilen) Die Studierenden ordnen die neuen Inhalte in den Gesamtkontext des entsprechenden Themengebietes ein und präsentieren fundierte Stellungnahmen zur Qualität der vorgestellten Ergebnisse. Erschaffen Die Studierenden können neue Anwendungsgebiete identifizieren und bekannte Verfahren verbessern oder mit neuen Ideen bekannte Probleme besser lösen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!

9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 886484	**Seminar: Ausgewählte Kapitel aus "The Art of Computer Programming"**	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Oliver Keszöcze
5	**Inhalt**	Donald E. Knuth ist vielen Studierenden der naturwissenschaftlichen und technischen Fächer hauptsächlich durch sein Textsatzsystem LaTeX bekannt. Sein wissenschaftliches Hauptwerk ist jedoch das Buch "The Art of Computer Programming" (TAOCP). Anders als der Titel suggeriert, handelt es sich bei dem Buch um kein Programmierhandbuch, sondern vielmehr um ein allumfassendes Kompendium von Algorithmen und Datenstrukturen. Es wird zu Recht als "die Bibel aller fundamentalen Algorithmen" bezeichnet. Insbesondere interessant ist, dass Knuth durch seine Herangehensweise bzw. Darstellung einer völlig neue Sichtweisen auf schon bekannte Probleme erlaubt. Dieses Seminar widmet sich ausgesuchten Kapiteln aus TAOCP. Teilnehmer setzen sich mit den Inhalten auseinander und stellen diese in einem Vortrag und einer schriftlichen Ausarbeitung vor. Die Ausarbeitung kann einen Programmieranteil enthalten. Nach Rücksprache können auch eigene Kapitel ausgesucht werden. Auf Wunsch kann der Vortrag und/oder die Ausarbeitung auf Englisch verfasst werden.
6	**Lernziele und Kompetenzen**	Fachkompetenz Verstehen Die Studierenden tragen grundlegende Inhalte aus dem Buch "The Art of Computer Programming" vor. Anwenden Die Studierenden können eigenständig neue Beispiele und Anwendungsszenarien aufstellen. Evaluieren (Beurteilen) Die Studierenden können die vorgestellten Inhalte in einen wissenschaftlichen Kontext setzen und ihre Güte beurteilen. Lern- bzw. Methodenkompetenz Die Studierenden arbeiten sich selbstständig in einen Abschnitt aus "The Art of Computer Programming" ein und suchen hierbei selbstständig nach verwandten Arbeiten, um den Kontext zu verstehen und aufzuarbeiten. Die Studierenden lernen, wissenschaftliche Inhalt mündlich sowie schriftlich leicht verständlich darzustellen. Selbstkompetenz

		Die Studierenden können eigene Stärken und Schwächen im Bereich ihrer Präsentationstechniken reflektieren und die eigene Entwicklung planen. Sozialkompetenz Die Studierenden beteiligen sich aktiv an den Vorträgen der jeweils anderen Studierenden durch fachbezogene Fragen zum Thema wie auch Rückmeldung zu Vortragsstil.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 153330	**Seminar Einführung in die Kryptografie** Seminar: Introduction to cryptography	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Dr.-Ing. Marc Spisländer
5	**Inhalt**	In diesem Seminar werden symmetrische und asymmetrische Verfahren zur Identifikation, Verschlüsselung und Signierung behandelt. Dazu werden sowohl die mathematischen Grundlagen vermittelt als auch die entsprechenden Algorithmen vorgestellt.
6	**Lernziele und Kompetenzen**	Die Studierenden • erläutern wesentliche Konzepte der modernen Kryptografie; • klassifizieren Kryptoverfahren und charakterisieren deren Stärken und Schwächen; • tragen vor Publikum über wissenschaftliche Ergebnisse vor; • recherchieren selbständig Fachliteratur; • fassen wissenschaftliche Erkenntnisse in Schriftform zusammen; • nutzen Verbesserungshinweise des Betreuers zur Analyse eigener Stärken und Schwächen und leiten daraus Konsequenzen für ihr künftiges Lern-Handeln ab; • können komplexe fachbezogene Inhalte klar und zielgruppengerecht präsentieren und argumentativ vertreten.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93656	**Seminar Energieinformatik** Seminar: Energy informatics	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Marco Pruckner
5	**Inhalt**	Teilnehmende arbeiten sich selbständig anhand der ausgewählten wissenschaftlichen Literatur in ein vorgeschlagenes Thema aus dem Bereich der Energieinformatik ein, erstellen dazu eine schriftliche Ausarbeitung, bereiten einen Seminarvortrag vor und präsentieren ihn vor Lehrenden, interessierten wissenschaftlichen Mitarbeitern des Instituts und Kommilitonen der Lehrveranstaltung.
6	**Lernziele und Kompetenzen**	Zu den zu erwerbenden Kompetenzen zählen • die Literaturrecherche, • korrektes Zitieren, • die Fähigkeit auszuwählen, welche Aspekte in der Ausarbeitung und im Vortrag behandelt werden, • zielgruppengerechtes Schreiben, • die Verwendung von Textverarbeitungswerkzeugen für die Präsentation und die Ausarbeitung, • sicheres Auftreten beim Vortragen eines wissenschaftlichen Themas sowie • fachspezifische Fragen zum ausgewählten Themengebiet zu beantworten. Am Ende jeder Lehreinheit • stellen die Teilnehmenden gezielt Fragen zum vorgetragenen Thema, • diskutieren über fachliche Aspekte der Präsentation, • üben konstruktive Kritik an der Darstellung des Themas und • bewerten die Präsentation aus vortragstechnischer Sicht.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• gesammelte Bücher, Artikel und Weblinks zu einem einschlägigen Themengebiet • Marco Pruckner. Allgemeine Hinweise zu Seminarvorträgen und -ausarbeitungen.

1	**Modulbezeichnung** 319374	**Seminar 'Hallo Welt!' für Fortgeschrittene** Seminar: "Hello World!" for advanced students	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten. Es besteht eine Anwesenheitspflicht.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Daniela Novac
5	**Inhalt**	Programmierwettbewerbe wie der **International Collegiate Programming Contest (ICPC)** der ACM bieten die Möglichkeit, die eigenen Programmier- und Teamfähigkeiten an einer Vielzahl algorithmischer Probleme aus ganz verschiedenen Gebieten wie Geometrie, Kombinatorik, String-Verarbeitung und Zahlentheorie zu testen. Dabei treten die Studenten in 3er-Teams an, haben aber nur einen Computer zur Verfügung. Oft ist die Teamstrategie entscheidend für den Erfolg der Gruppe. In diesem Seminar werden wichtige Algorithmen zur Lösung von Problemen aus den verschiedenen Gebieten in wöchentlichen, studentischen Vorträgen vorgestellt und Standardverfahren eingeübt. Neben den Vorträgen werden die aktuell zu lösenden Aufgaben in einer simulierten Wettbewerbssituation in 3er-Teams besprochen und Lösungsansätze in der Gesamtgruppe diskutiert. Das Seminar bereitet auf die Teilnahme am Programmierwettbewerb der Universität Erlangen-Nürnberg Ende des Sommersemesters vor. Es besteht Teilnahmepflicht für diesen Wettbewerb.
6	**Lernziele und Kompetenzen**	Nach der Teilnahme an diesem Seminar sind die Studierenden in der Lage: • erforderliche Literatur aufzufinden, zu analysieren und zu bewerten, • sich eigenständig in ein Themengebiet einzuarbeiten, • Grundzüge der Präsentationstechniken anzuwenden, • eine Präsentation mit Begleitmaterial für ein Fachpublikum zu entwickeln, • einen Vortrag im vorgegebenen Zeitrahmen durchzuführen, • Sachverhalte unter Fachleuten zu diskutieren, • grundlegendende und fortgeschrittene Algorithmen aus diversen Themengebieten darzulegen, • den Zusammenhang zwischen der Laufzeit und dem Speicherverbrauch von Programmen und der theoretischen Komplexität (O-Kalkül) anzugeben, • algorithmische Probleme zu analysieren und die gelernten Algorithmen zur Lösung anzuwenden, • Lösungsideen in Kleingruppen zu entwickeln und diese in der Gesamtgruppe zu präsentieren.
7	**Voraussetzungen für die Teilnahme**	AuD (oder GdP), Spaß am Programmieren, Teilnahme am ICPC

8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Skiena/Revilla, Programming Challenges. The Programming Contest Training Manual. Springer 2003. Cormen/Leiserson/Rivest/Stein, Introduction to Algorithms. MIT Press 2001.

1	**Modulbezeichnung** 95790	**Seminar Koalgebraische Logik** Seminar: Coalgebraic logic	**5 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Gabriele Schönberger Prof. Dr. Lutz Schröder
5	**Inhalt**	• Basic categorical notions • Coalgebraic logic via predicate liftings • Separation and expressivity • Soundness and completeness • Algebraic semantics • Complexity of reasoning • Modular construction of logics • Coalgebraic hybrid logic • Reasoning with global assumptions • Fixpoint logics • Coalgebraic automata theory • Fuzzy coalgebraic logic • Coalgebraic predicate logic
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben grundlegende Definitionen und Resultate zu Koalgebra, Modallogik, und koalgebraischer Logik wieder und ordnen Beispiele korrekt ein. Verstehen Die Studierenden erläutern generische Schlussfolgerungsverfahren und algorithmische Verfahren in der koalgebrischen Logik und instanziieren sie auf konkrete Logiken. Anwenden Die Studierenden übertragen koalgebraische generische Verfahren selbständig auf Beispiele. Erschaffen Die Studierenden bereiten Resultate des Gebiets und ihre Herleitung eigenständig zur Präsentation auf. Lern- bzw. Methodenkompetenz Die Studierenden erarbeiten sich selbständig den Inhalt wissenschaftlicher Veröffentlichungen und geben diesen in einem strukturierten Vortrag verständlich wieder. Sozialkompetenz Die Studierenden arbeiten in einer größeren Gruppe an komplexen Problemen und erläutern eigene Einsichten den Mitstudierenden in verständlicher Form.
7	**Voraussetzungen für die Teilnahme**	Empfohlen werden die vorherige Teilnahme an "Grundlagen der Logik in der Informatik" sowie an einer Vertiefungsveranstaltung in der theoretischen Informatik
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung Die Seminarleistung besteht in einem Vortrag sowie, abhängig von der Vortragslänge, einer schriftlichen Ausarbeitung

11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	(The following list is, while not exhaustive, meant mainly as a fairly broad guide to the literature, and not to be understood as a mandatory reading list.) • Lawrence S. Moss: Coalgebraic Logic. Ann. Pure Appl. Logic 96(1-3): 277-317 (1999) • Dirk Pattinson: Expressive Logics for Coalgebras via Terminal Sequence Induction. Notre Dame Journal of Formal Logic 45(1): 19-33 (2004) • Dirk Pattinson: Coalgebraic modal logic: soundness, completeness and decidability of local consequence. Theor. Comput. Sci. 309(1-3): 177-193 (2003) • Lutz Schröder: A finite model construction for coalgebraic modal logic. J. Log. Algebr. Program. 73(1-2): 97-110 (2007) • Lutz Schröder, Dirk Pattinson: Shallow Models for Non-iterative Modal Logics. KI 2008: 324-331 • Dirk Pattinson, Lutz Schröder: Beyond Rank 1: Algebraic Semantics and Finite Models for Coalgebraic Logics. FoSSaCS 2008: 66-80 • Lutz Schröder: Expressivity of coalgebraic modal logic: The limits and beyond. Theor. Comput. Sci. 390(2-3): 230-247 (2008) • Lutz Schröder, Dirk Pattinson: Strong Completeness of Coalgebraic Modal Logics. STACS 2009: 673-684 • Robert S. R. Myers, Dirk Pattinson, Lutz Schröder: Coalgebraic Hybrid Logic. FOSSACS 2009: 137-151 • Lutz Schröder, Dirk Pattinson, Clemens Kupke: Nominals for Everyone. IJCAI 2009: 917-922 • Lutz Schröder, Yde Venema: Flat Coalgebraic Fixed Point Logics. CONCUR 2010: 524-538 • Lutz Schröder, Dirk Pattinson: Description Logics and Fuzzy Probability. IJCAI 2011: 1075-1081 • Tadeusz Litak, Dirk Pattinson, Katsuhiko Sano, Lutz Schröder: Coalgebraic Predicate Logic. ICALP (2) 2012: 299-311 • Daniel Gorín, Lutz Schröder: Subsumption Checking in Conjunctive Coalgebraic Fixpoint Logics. Advances in Modal Logic 2014: 254-273

		• Daniel Gorín, Dirk Pattinson, Lutz Schröder, Florian Widmann, Thorsten Wißmann: COOL - A Generic Reasoner for Coalgebraic Hybrid Logics (System Description). IJCAR 2014: 396-402 • Corina Cîrstea, Clemens Kupke, Dirk Pattinson: EXPTIME Tableaux for the Coalgebraic mu-Calculus. Logical Methods in Computer Science 7(3) (2011) • Yde Venema: Automata and fixed point logic: A coalgebraic perspective. Inf. Comput. 204(4): 637-678 (2006) • Clemens Kupke, Yde Venema: Coalgebraic Automata Theory: Basic Results. Logical Methods in Computer Science 4(4) (2008) • Stefan Milius, Dirk Pattinson, Lutz Schröder: Generic Trace Semantics and Graded Monads. CALCO 2015: 253-269 • Clemens Kupke, Dirk Pattinson, Lutz Schröder: Reasoning with Global Assumptions in Arithmetic Modal Logics. FCT 2015: 367-380 • Lutz Schröder, Dirk Pattinson, Tadeusz Litak: A Van Benthem/ Rosen theorem for coalgebraic predicate logic. J. Log. Comput. 27(3): 749-773 (2017) • Daniel Hausmann, Lutz Schröder: Optimal Satisfiability Checking for Arithmetic \mu -Calculi. FoSSaCS 2019: 277-294 • Daniel Hausmann, Lutz Schröder: Game-Based Local Model Checking for the Coalgebraic mu-Calculus. CONCUR 2019: 35:1-35:16

1	**Modulbezeichnung** 609624	**Seminar Kommunikationssysteme (B.Sc.)** Seminar communication systems (B.Sc.)	**5 ECTS**
2	Lehrveranstaltungen	Seminar: Seminar: Kommunikationssysteme (Seminar on Cybersecurity in Smart Grids) (2 SWS)	5 ECTS
3	Lehrende	Prof. Dr. Reinhard German	

4	**Modulverantwortliche/r**	Prof. Dr. Reinhard German
5	**Inhalt**	Teilnehmende arbeiten sich selbständig anhand der ausgewählten wissenschaftlichen Literatur in ein vorgeschlagenes Thema aus dem Bereich der Kommunikationssysteme ein, erstellen dazu eine schriftliche Ausarbeitung, bereiten einen Seminarvortrag vor und präsentieren ihn vor Lehrenden, interessierten wissenschaftlichen Mitarbeitern des Instituts und Kommilitonen der Lehrveranstaltung.
6	**Lernziele und Kompetenzen**	Zu den zu erwerbenden Kompetenzen zählen • die Literaturrecherche, • korrektes Zitieren, • die Fähigkeit auszuwählen, welche Aspekte in der Ausarbeitung und im Vortrag behandelt werden, • zielgruppengerechtes Schreiben, • die Verwendung von Textverarbeitungswerkzeugen für die Präsentation - und die Ausarbeitung, • sicheres Auftreten beim Vortragen eines wissenschaftlichen Themas sowie • fachspezifische Fragen zum ausgewählten Themengebiet zu beantworten. Am Ende jeder Lehreinheit • stellen die Teilnehmenden gezielt Fragen zum vorgetragenen Thema, • diskutieren über fachliche Aspekte der Präsentation, • üben konstruktive Kritik an der Darstellung des Themas und • bewerten die Präsentation aus vortragstechnischer Sicht.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung Prüfungsleistung, Seminarleistung, benotet, 5.0 ECTS Anteil an der Berechnung der Modulnote: 100.0 % Der Scheinerwerb erfolgt durch einen 45-minütigen Vortrag.
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 903776	**Seminar Machine Learning and Data Analytics for Industry 4.0** Seminar machine learning and data analytics for industry 4.0	**5 ECTS**
2	Lehrveranstaltungen	Hauptseminar: Maschinelles Lernen und Datenanalytik für Industrie 4.0 (2 SWS) Attendance of all meetings is required.	5 ECTS
3	Lehrende	Johannes Roider Prof. Dr. Björn Eskofier	

4	**Modulverantwortliche/r**	Prof. Dr. Björn Eskofier An Nguyen Johannes Roider
5	**Inhalt**	Companies in all kinds of industries are producing and collecting rapidly more and more data from various sources. This is enabled by technologies such as the Internet of Things (IoT), Cyber-physical systems (CPS) and cloud computing. Hence, there is an increasing demand in industry and research for students and graduates with machine learning and data analytics skills in the Industry 4.0 context. The goal of this seminar is to give students insights about state-of-the-art machine learning and data analytics methods for industrial and business applications. In this seminar, the Industry 4.0 term will not only be limited to manufacturing processes, but comprise all business functions. Students will mainly work independently on either an implementation-centric or a research-centric topic. The implementation-centric topics will focus primarily on the implementation of algorithms and analytical components (using provided or open source datasets), while the research-centric topics will focus on researching and structuring literature of a specific field of interest. Several topics will be provided, but students are also encouraged to propose their own topics when applying for the seminar. In the regular meetings, students will learn about fundamentals and trends in Industry 4.0 from a machine learning perspective, common machine learning techniques and their implementation, project management of data analytics projects in businesses, as well as best practices for presentations and scientific work. The programme will be complemented by talks from invited experts in the domain. Furthermore, students will present results from literature research and data analytics projects.
6	**Lernziele und Kompetenzen**	• Students will develop an understanding of the current hot field of machine learning and data analytics in businesses • Students will learn to research and present a topic within the context of machine learning and data analytics in businesses independently • Students will learn to identify opportunities, challenges and limitations of corresponding ML approaches in businesses

		• Students will develop the skill to identify and understand relevant literature and to present their findings in a structured manner • Students will learn to present implementation and validation results in form of a demonstration and/or report
7	**Voraussetzungen für die Teilnahme**	• Prior knowledge of machine learning via courses like Pattern Analysis, Pattern Recognition, Deep Learning, Machine Learning for Time Series, or equivalent is expected. Alternatively, first data science project experience, for example as working student in a company, can be sufficient. • Motivation to explore scientific findings (e.g. via literature research) • Motivation to code and analyze data
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%) • 50% of grade: Presentation (20 minutes) • 50% of grade: 4 pages IEEE standard paper (excluding references) (+ code submission)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Lei, Yaguo, Naipeng Li, Liang Guo, Ningbo Li, Tao Yan, and Jing Lin. "Machinery Health Prognostics: A Systematic Review from Data Acquisition to RUL Prediction. Mechanical Systems and Signal Processing 104 (May 2018): 799834.https://doi.org/10.1016/j.ymssp.2017.11.016. • Rojas, Eric, Jorge Munoz-Gama, Marcos Sepúlveda, and Daniel Capurro. "Process Mining in Healthcare: A Literature Review. Journal of Biomedical Informatics 61 (June 1, 2016): 22436. https://doi.org/10.1016/j.jbi.2016.04.007. • Wil M. P. van der Aalst. Process Mining: Data Science in Action 2nd edition, Springer 2016. ISBN 978-3-662-49851-4 • Wang, Lihui, and Xi Vincent Wang. Cloud-Based Cyber-Physical Systems in Manufacturing. Cham: Springer International Publishing, 2018. https://doi.org/10.1007/978-3-319-67693-7.

1	**Modulbezeichnung** 349413	**Seminar Sprachtechnologie für Sprachpathologien** Speech technologies for speech pathologies	**5 ECTS**
2	Lehrveranstaltungen	Hauptseminar: Seminar Automatic Analysis of Voice, Speech and Language Disorders in Speech Pathologies (4 SWS)	5 ECTS
3	Lehrende	Prof. Seung Hee Yang Prof. Dr.-Ing. Andreas Maier	

4	**Modulverantwortliche/r**	Prof. Seung Hee Yang
5	**Inhalt**	This seminar deals with how the diagnosis and therapy of different speech pathologies can be supported by speech technology. The participants should present selected speech, speech and voice disorders in a lecture and demonstrate corresponding technologies in the field of pattern recognition and speech processing.
6	**Lernziele und Kompetenzen**	Die Studierenden • erlernen die Literaturrecherche. • arbeiten sich selbstständig anhand der gefundenen Literatur in die Thematik der automatischen Analyse von Sprach-, Sprech- und Stimmstörungen ein. • wählen einen Schwerpunkt und bereiten diesen im Rahmen einer Präsentation so auf, dass er für andere Teilnehmer des Seminars verständlich ist. • lernen die Anforderungen an einen wissenschaftlichen Vortrag auf einer internationalen Konferenz kennen. • halten einen Vortrag in der international üblichen Fachsprache Englisch (davon ausgenommen sind Studierende aus dem Ausland, die in Deutschland studieren, um Deutsch zu lernen)
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 716001	**Seminar SystemC** Seminar: SystemC	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Joachim Falk
5	**Inhalt**	Typischerweise werden Hardware und Software eines eingebetteten Systems separat in verschiedenen Programmiersprachen entwickelt. Fortschrittliche Methoden vereinen die Entwicklung beider Welten in einem gemeinsamen Entwurfsfluss. Dies vereinfacht Integration, Simulation und Verifikation des gesamten Systems. Die Systembeschreibungssprache SystemC verfolgt diesen modernen Ansatz und findet zunehmend Akzeptanz in Industrie und Forschung.
6	**Lernziele und Kompetenzen**	Dieses Seminar vermittelt auf praxisnahe Weise nötige Grundlagen und Methoden für die Entwicklung eingebetteter Systeme in SystemC. Dabei behandeln wir im Einzelnen: • Grundlagen des Hardware/Software-Entwurf • Einführung in C++ • Einführung in SystemC • Entwurf eingebetteter Systeme in SystemC am Beispiel einer interaktiven Fraktaldarstellungsapplikation. • Platformsimulation eines SoC mittels SystemC-TLM und OVP, einem Instruktionssatz-Simulator von Imperas. Fachkompetenz - Verstehen • Die Studierenden erläutern die Grundlagen des Hardware/Software-Entwurfs. • Die Studierenden erläutern die verschiedenen Modellierungsebenen für den Hardware/Software-Entwurf. • Die Studierenden zeigen den Zusammenhang zwischen Simulationsgeschwindigkeit und Modellierungsebenen auf. Fachkompetenz - Anwenden • Die Studierenden benutzen die C++ Klassenbibliothek SystemC zur Modellierung der Hardwarekomponenten und SystemC-TLM zur Modellierung der Busstrukturen Ihres MPSoC Hardware/Software-Entwurfs. • Die Studierenden benutzen Instruktionssatz-Simulatoren zur Simulation der Softwarekomponenten Ihres MPSoC Hardware/Software-Entwurfs. Fachkompetenz - Analysieren • Die Studierenden erforschen durch Analyse und Literaturrecherche die Vor- und Nachteile verschiedener Simulationsverfahren auf verschiedenen Modellierungsebenen. Fachkompetenz - Erschaffen • Die Studierenden erstellen einen MPSoC Hardware/Software-Entwurf einer Fraktaldarstellungsapplikation in SystemC. Lern- bzw. Methodenkompetenz

		• Die Studierenden arbeiten sich selbstständig in eine wissenschaftliche Veröffentlichungen ein und nutzen das erworbene Wissen aus dem praktischen Teil, um den Kontext der Veröffentlichung zu verstehen und aufzubereiten. Sozialkompetenz • Die Studierenden beteiligen sich aktiv an den Vorträgen der jeweils anderen Studierenden durch fachbezogene Fragen zum Thema wie auch Rückmeldung zu Vortragsstil.
7	**Voraussetzungen für die Teilnahme**	Die Auswahl dieses Moduls schließt die Auswahl der Module „SystemC als Praktikum“ und „SystemC als Seminar für IuK“ aus. Wenn Sie I&K studieren, dann wählen Sie bitte entweder das Modul „SystemC als Praktikum“ oder „SystemC als Seminar für IuK“, um Ihr 2.5 ECTS Praktikum oder Seminar abzudecken.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/praktika/praktikum-systemc/

1	**Modulbezeichnung** 863761	**Seminar Theoretische Informatik** Theoretical computer science (seminar)	**5 ECTS**
2	Lehrveranstaltungen	Hauptseminar: Seminar Theoretische Informatik (2 SWS)	-
3	Lehrende	Prof. Dr. Lutz Schröder	

4	**Modulverantwortliche/r**	Prof. Dr. Lutz Schröder
5	**Inhalt**	• Theory of concurrency • Programming semantics • Categories in computer science • Logic in computer science • Theory of artificial intelligence
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben grundlegende Definitionen und Resultate aus einem vertieften Bereich der Theoretischen Informatik wieder. Verstehen Die Studierenden erläutern grundlegende algorithmische und deduktive Verfahren. Anwenden Die Studierenden erläutern Anwendungen theoretischer Methoden in konkreten informatischen Kontexten. Erschaffen Die Studierenden bereiten Resultate des Gebiets und ihre Herleitung eigenständig zur Präsentation auf. Lern- bzw. Methodenkompetenz Die Studierenden erarbeiten sich selbständig den Inhalt wissenschaftlicher Veröffentlichungen und geben diesen in einem strukturierten Vortrag verständlich wieder. Sozialkompetenz Die Studierenden arbeiten in einer größeren Gruppe an komplexen Problemen und erläutern eigene Einsichten den Mitstudierenden in verständlicher Form.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 635405	**Seminar Wissensrepräsentation und -verarbeitung** Knowledge representation and -processing	**5 ECTS**
2	Lehrveranstaltungen	Hauptseminar: Seminar Wissensrepräsentation und -verarbeitung (2 SWS)	-
3	Lehrende	Prof. Dr. Michael Kohlhase PD Dr. Florian Rabe	

4	**Modulverantwortliche/r**	Prof. Dr. Michael Kohlhase PD Dr. Florian Rabe
5	**Inhalt**	Dieses Seminar behandelt ausgewählte Themen aus dem Gebiet der Wissensrepräsentation und -Verarbeitung, mit einem Fokus auf mathematisches Wissen. Die Vortragsthemen sind sehr unterschiedlich im Schwierigkeitsgrad, sie reichen von Einführungsthemen für ambitionierte Bachelor-Studenten bis zur Forschungsfront für Doktoranden. Sie spiegeln recht gut die Forschungsinteressen der KWARC Gruppe wieder. Daher ist dieses Seminar sehr gut geeignet um in die Arbeitsgruppe einzusteigen (z.B. für eine Promotion).
6	**Lernziele und Kompetenzen**	Aktuelle Fachliteratur verstehen, Forschungsthemen für ein Fachpublikum verständlich aufbereiten und vortragen, Kommunikation mit Experten. Fachkompetenz Wissen Die Studierenden geben grundlegende Definitionen und Resultate aus einem vertieften Bereich der Wissensrepräsentation und -Verarbeitung wieder. Verstehen Die Studierenden erläutern grundlegende algorithmische und deduktive Verfahren. Sie tragen durch Literaturstudium erworbenes Wissen vor und erläutern es einem kleinem Fachpublikum aus Mitstudierenden. Anwenden Die Studierenden erläutern Anwendungen theoretischer Methoden in konkreten informatischen Kontexten. Erschaffen Die Studierenden bereiten Resultate des Gebiets und ihre Herleitung eigenständig zur Präsentation auf. Lern- bzw. Methodenkompetenz Die Studierenden erarbeiten sich selbständig den Inhalt wissenschaftlicher Veröffentlichungen und geben diesen in einem strukturierten Vortrag verständlich wieder. Sozialkompetenz Die Studierenden arbeiten in einer größeren Gruppe an komplexen Problemen und erläutern eigene Einsichten den Mitstudierenden in verständlicher Form.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!

9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	Wird in der Veranstaltung im persönlichen Betreuungsgespräch zwischen Lehrenden und Studierenden gemeinsam für den Seminarvortrag ausgewählt.

1	**Modulbezeichnung** 834345	**Systems- and Networks-on-a-Chip für INF** Systems- and networks-on-a-chip for computer scientists	**5 ECTS**
2	Lehrveranstaltungen	Seminar: Systems- and Networks-on-a-Chip (2 SWS)	5 ECTS
3	Lehrende	Paul Krüger Dr.-Ing. Stefan Wildermann	

4	**Modulverantwortliche/r**	Joachim Falk Dr.-Ing. Stefan Wildermann
5	**Inhalt**	Eingebettete Systeme spielen im alltäglichen Leben eine immer größere Rolle. Gleichzeitig nimmt die Komplexität dieser Systeme immer weiter zu. Durch die heutige Technologie ist es möglich, Millionen, in naher Zukunft Milliarden von Transistoren auf einem Chip zu platzieren. Dies führt dazu, dass häufig das komplette eingebettete System, ein sogenanntes System-on-a-Chip (SoC), auf einem einzigen Chip realisiert werden kann. Die Vorteile einer verbesserten Performanz, niedrigerem Energieverbrauch sowie sinkenden Kosten sind dabei unter anderem durch die Wiederverwendung bestehender Komponenten bedingt. Eine der Herausforderungen bestehender SoCs besteht darin, eine korrekte und zuverlässige Kommunikation zwischen den Komponenten herzustellen. Aus diesem Grund wird den Komponenten eine netzwerkartige Kommunikation zur Verfügung gestellt, wodurch sogenannte Networks-on-a-Chip (NoCs) entstehen. Dieses Seminar beschäftigt sich mit der Problematik von Design, Synthese und Analyse bestehender und zukünftiger Systems- und Networks-on-a-Chip. Hierbei soll vor allem die Vereinbarkeit verschiedener Anforderungen an das System wie Kosten, Platz- und Energieverbrauch oder Zuverlässigkeit in den verschiedenen Phasen der Entwicklung betrachtet werden.
6	**Lernziele und Kompetenzen**	**Fachkompetenz - Verstehen** • Die Studierenden tragen die wesentlichen Inhalte einer ausgewählten wissenschaftlichen Veröffentlichung auf dem Gebiet der MPSoCs vor. • Die Studierenden veranschaulichen den grundlegenden Kontext der Veröffentlichung sowie deren wesentliche Neuerungen. • Die Studierenden bereiten den Inhalt der ausgewählten wissenschaftlichen Veröffentlichung sowie benötigte Grundlagen in einer Ausarbeitung auf. **Lern- bzw. Methodenkompetenz** • Die Studierenden arbeiten sich selbstständig in eine wissenschaftliche Veröffentlichung ein und suchen hierbei selbstständig nach verwandten Arbeiten, um den Kontext der Veröffentlichung zu verstehen und aufzubereiten. **Sozialkompetenz** • Die Studierenden beteiligen sich aktiv an den Vorträgen der jeweils anderen Studierenden durch fachbezogene Fragen zum Thema wie auch Rückmeldung zu Vortragsstil.

7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%) Die Bewertung der Prüfungsleistung setzt sich zusammen aus 50% Seminarvortrag (ca. 30 Minuten Präsentation + ca. 15 Minuten Frage und Antwort) und 50% schriftlicher Ausarbeitung (14 Seiten Seminarbericht).
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 819238	**Themen der Kategorientheorie**	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	apl.Prof.Dr. Stefan Milius
5	**Inhalt**	Verschiedene Themen der Kategorientheorie werden in Form eines "Reading Course" von den Teilnehmern erarbeitet und vorgetragen. Die Themenauswahl kann dabei flexibel auf die Interessen der Teilnehmer zugeschnitten werden. Mögliche, teilweise an "Algebra des Programmierens" unmittelbar anschließende, Themen sind z.B.: • freie Konstruktionen, universelle Pfeile und adjungierte Funktoren • Äquivalenzfunktoren • Monaden: Eilenberg-Moore und Kleisli-Kategorien; Freie Monaden; Becks Satz • Kartesisch abgeschlossenene Kategorien • Vollständige Halbordnungen (cpos), Einbettungen/ Projektionen, Limes-Kolimes-Koinzidenz, Lösung rekursive Domaingleichungen • Kan Erweiterungen • (symmetrische) monoidale Kategorien • Faktorisierungsstrukturen
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben Definitionen und Ergebnisse aus der Kategorientheorie und ihren Anwendungen in der Informatik wieder. Verstehen Die Studierenden erläutern grundlegende Methoden und Beweise der Kategorientheorie. Anwenden Die Studierenden wenden die Konzepte und Beweismethoden an, um einfache Problemstellungen kategoriell zu beschreiben und entsprechende Aussagen zu beweisen. Erschaffen Die Studierenden bereiten Resultate des Gebiets und ihre Herleitung eigenständig zur Präsentation auf. Lern- bzw. Methodenkompetenz Die Studierenden erarbeiten sich selbständig den Inhalt von Fachbüchern und wissenschaftlichen Veröffentlichungen und geben diesen in einem strukturierten Vortrag verständlich wieder. Sozialkompetenz Die Studierenden arbeiten in einer größeren Gruppe an komplexen Problemen und erläutern eigene Einsichten den Mitstudierenden in verständlicher Form.

7	**Voraussetzungen für die Teilnahme**	Grundkenntnisse der Kategorientheorie.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	• S. Mac Lane: Categories for the Working Mathematician, 2nd edition, Springer-Verlag, 1998. • J. Adamek, H. Herrlich and G.E. Strecker: Abstract and Concrete Categories: The Joy of Cats, 2nd edition, Dover Publications, 2009. • S. Awodey: Category Theory, 2nd edition, Oxford University Press, 2011. • E. Riehl: Category Theory in Context, Dover Publications, 2016.

Vertiefungsrichtung Datenbanksysteme

1	**Modulbezeichnung** 681735	**Datenbanken in Rechnernetzen und Transaktionssysteme** Distributed databases and transaction systems	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Transaktionssysteme / Transaction Systems (2 SWS) Vorlesung: Datenbanken in Rechnernetzen (2 SWS)	- -
3	Lehrende	Prof. Dr.-Ing. Richard Lenz	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Richard Lenz
5	**Inhalt**	**DBRN:** Rechnernetze entsprechen dem momentanen Stand der Technik; isolierte Rechnersysteme nehmen an Zahl und Bedeutung ab. Das Betreiben von Datenbanksystemen in Rechnernetzen erfordert neuartige Konzepte, die über die einer zentralisierten Datenbankverwaltung hinausgehen. In der Vorlesung werden Ansätze zur Datenbankverwaltung in verteilten Systemen vorgestellt. Verteilte Datenbanken, Parallele Datenbanken, DB-Sharing und heterogene Datenbanksysteme werden untersucht. Darüber hinaus widmet sich ein weiteres Hauptkapitel der Vorlesung der Verwendung und dem Betrieb von Datenbanksystemen im Internet. **TAS:** Transactions are the core mechanism to guarantee database consistency in the presence of failures. The lecture introduces the cornerstones of the Transaction Concept and related techniques and system architectures. Topics are: • Reconstructing the Transaction Model • Advanced Transaction Models • Queued transaction processing • Implementing the ACID properties of transactions: Concurrency control, logging and recovery • TP Monitors: TRPC, Architecture of TP Monitor, Transaction Manager This course generalizes the transaction concept from its traditional database system domain to the broader context of client-server computing. The course begins by defining basic terminology and concepts. The role of a transaction processing system in application design, implementation, and operation is covered. Subsequent lectures cover the theory and practice of implementing locking, logging, and the more generic topic of implementing transactional resource managers.
6	**Lernziele und Kompetenzen**	**DBRN:** Die Studierenden • Benennen Ziele verteilter Datenhaltungssysteme • Erklären verschiedene Zielkonflikte, insbesondere das CAP-Theorem • Unterscheiden verschiedene Varianten verteilter Datenhaltungssysteme

- Erläutern die Optionen zur Metadatenverwaltung in verteilten Datenbanken
- Definieren horizontale und vertikale Fragmentierungen für relationale Datenbanken
- Erklären die Transformationsschritte und Optimierungen der verteilten Anfrageverarbeitung anhand konkreter Beispiele
- Erklären Algorithmen zur verteilten Ausführung von Verbund-Operationen
- Erläutern die Problematik der Deadlock-Erkennung bei verteilten Sperrverfahren
- Unterscheiden die Funktionsweise von Sperrverfahren, Zeitstempelverfahren und Optimistischen Verfahren zur Synchronisation verteilter Transaktionen
- Benennen und erklären verschiedene Verfahren zur Replikationskontrolle
- Erläutern Techniken und Verfahren zur Abschwächung der Konsistenzanforderungen an replizierte Datenbestände
- Erklären die Funktionsweise hochskalierbarer No-SQL Datenbanken am Beispiel der Replikationsmechanismen im Datenbanksystem Cassandra
- Unterscheiden und erläutern Realisierungsalternativen zur Kopplung und Integration heterogener autonomer Datenbanken
- Erläutern die erweiterte Schema-Architektur für föderative Datenbanksysteme
- Erklären die Abbildungsvarianten GaV und LaV für die Implementierung Föderativer Datenbanken

TAS:

Die Studierenden

- Erklären die Zielsetzungen und Grenzen transaktionaler Systeme
- Unterscheiden verschiedene erweiterte Transaktionsmodelle
- Erläutern wie die Verfügbarkeit verteilter transaktionaler Systeme durch Queued Transactions" verbessert werden kann
- Erklären typische Nebenläufigkeitsanomalien
- Erläutern mit konkreten Beispielen was Wiederherstellbarkeit und Striktheit bedeuten
- Erklären Ziele und Funktionsweise von Sperrverfahren, hierarchischen Sperrverfahren und zusätzlichen Sperrmodi
- Erläutern Isolationsstufen zur Abschwächung des Synchronisationsaufwands
- Erklären die grundlegenden Aufgaben und Funktionen eines Recovery Managers"
- Unterscheiden verschiedene Klassen von Wiederherstellungsalgorithmen
- Erklären Zweck und Funktionsweise von Checkpoints" und Fuzzy Checkpoints"
- Erklären im Detail wie das Zwei-Phasen Freigabeprotokoll funktioniert

		• Erläutern Ziele und Funktionsweise des Drei-Phasen-Freigabeprotokolls und Paxos-Commit • Erläutern die Funktionsweise verteilter Transaktionssysteme auf der Basis der standardisierten Schnittstellen in X-Open/ DTP
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Datenbanksysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	siehe Lehrveranstaltungen

1	**Modulbezeichnung** 710850	**eBusiness Technologies und Evolutionäre Informationssysteme** eBusiness technologies and evolutionary information systems	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Richard Lenz
5	**Inhalt**	**EBT:** • Überblick und Einblick in die wichtigsten Themen des Bereichs Business • User Interface, Business Logic und Database Layer • Agile Softwareentwicklung • Integration von Enterprise-Applikationen • Cloud & Container • DevOps **EIS:** • Grundlagen rechnergestützter Informationssysteme und organisatorisches Lernen • Erfolgsfaktoren für Projekte • Software Wartung vs. Software Evolution • Architekturmodelle • Grundprinzipien evolutionärer Systeme • Datenqualität in Informationssystemen **Contents:** **EBT:** • Modern technologies to implement Web-Applications for eBusiness • User Interface, Business Logic and Database Layer • Agile Software Development • Integration of Enterprise-Applications • Cloud & Container • DevOps **EIS:** • IT-Support for Organizational Learning • Success- and Failure Factors for large scale IT-Projects • Software Maintenance vs. Software Evolution • Architectural Styles and their Impact on Evolvability • Principles for Evolvable Systems • Data Quality in Information Systems
6	**Lernziele und Kompetenzen**	**EBT:** Die Studierenden • identifizieren die wichtigsten Themen des Bereichs eBusiness, von den Anwendungen bis zu den Implementierungen • verstehen Zusammenhänge der B2B-Integration und der Realisierung von eBusiness-Anwendungen • wiederholen Grundlagen des Webs

- vergleichen technische Eigenschaften von HTTP-, Web- und Application Servern
- vergleichen Markup Languages (HTML, XML)
- unterscheiden Ansätze zur Schema-Modellierung wie DTD und XML Schema und erkennen die unterschiedliche Leistungsfähigkeit
- verstehen Methoden zur evolutionsfähigen Gestaltung von Datenstrukturen in XML
- unterscheiden Vorgehen bei der Datenhaltung und verschiedene Ansätze für den Datenbankzugriff
- verstehen Objekt-relationale Mapping Frameworks am Beispiel von Hibernate und JPA
- verstehen Komponentenmodelle wie Enterprise JavaBeans (EJB) aus dem JEE Framework
- unterscheiden das EJB Komponentenmodell von den OSGi Bundles und den Spring Beans
- verstehen und unterscheiden grundlegende Web Service Techniken wie SOAP und WSDL
- unterscheiden Herangehensweisen zur dynamischen Generierung von Webseiten
- verstehen grundlegende Eigenschaften eines Java-basierten Front-End-Frameworks am Beispiel von JSF
- verstehen grundlegende Eigenschaften von Service-orientierten Architekturen (SOA)
- verstehen agile Vorgehensmodelle zur Software-Entwicklung am Beispiel von Scrum
- unterscheiden agile Verfahren wie Scrum von iterativ-inkerementellen Verfahren wie RUP
- verstehen die Wichtigkeit von Code-Beispielen um die praktische Anwendbarkeit des theoretischen Wissens zu veranschaulichen.
- können die Code-Beispiele eigenständig zur Ausführung bringen und die praktischen Erfahrungen interpretieren und bewerten
- gestalten eigene Lernprozesse selbständig.
- schätzen ihre eigenen Stärken und Schwächen im Hinblick auf die unterschiedlichen Architektur-Schichten ein(Benutzerinteraktion, Applikationslogik, Schnittstellenintegration, Datenbanksysteme)
- identifizieren eine eigene Vorstellung als zukünftige Software-Architekten und können die eigene Entwicklung planen
- reflektieren durch regelmäßige fachbezogene Fragen des Dozenten Ihre eigene Lösungskompetenz.

EIS:

Die Studierenden:

- definieren die Begriffe "Informationssysteme", "evolutionäre Informationssyste" und "organisatorisches Lernen"
- grenzen die Begriffe "Wissen" und "Information" gegeneinander ab

		• charakterisieren die in der Vorlesung erläuterten Formen der organisatorischen Veränderung • erklären das SEKI Modell nach Nonaka und Takeuchi • nennen Beispiele für die in der Vorlesung behandelten Formen der Wissensrepräsentation in IT-Systemen • nennen typische Erfolgs- und Risikofaktoren für große IT-Projekte • erklären die Kraftfeldtheorie nach Kurt Lewin • unterscheiden Typen von Software gemäß der Klassifikation nach Lehman und Belady • unterscheiden die in der Vorlesung vorgestellten Arten der Software Wartung • benennen die Gesetzmäßigkeiten der Software-Evolution nach Lehman und Belady • bewerten die in der Vorlesung vorgestellten Vorgehensmodelle zur Softwareerstellung im Kontext der E-Typ-Software • nennen die in der Vorlesung vorgestellten Aspekte der Evolutionsfähigkeit von Software • erklären, wie die in der Vorlesung vorgestellten Methoden zur Trennung von Belangen beitragen • erklären das Konzept des "Verzögerten Entwurfs" • erklären die Vor- und Nachteile generischer Datenbankschemata am Beispiel von EAV und EAV/CR • charakterisieren die in der Vorlesung vorgestelten Architekurkonzepte • grenzen die in der Vorlesung vorgestellten Integrationsanforderungen gegeneinander ab • erklären wie Standards zur Systemintegration beitragen und wo die Grenzen der Standardisierung liegen • erklären das Prinzip eines Kommunikationsservers und der nachrichtenbasierten Integration • erklären den Begriff "Prozessintegration" • definieren den Begriff "Enterprise Application Integration" (EAI) • unterscheiden die in der Vorlesung vorgestellten Integrationsansätze • erklären die in der Vorlesung vorgestellten Dimensionen der Datenqualität • unterscheiden die grundlegenden Messmethoden für Datenqualität • erklären das Maßnahmenportfolio zur Verbesserung der Datenqualität nach Redman • benennen die in der Vorlesung vorgestellten Methoden zur Verbesserung der Datenqualität
7	**Voraussetzungen für die Teilnahme**	Programmieren in Java, Datenbanken (SQL)
8	**Einpassung in Studienverlaufsplan**	Semester: 5

9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Datenbanksysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	siehe Lehrveranstaltungsbeschreibungen

1	**Modulbezeichnung** 43961	**Knowledge Discovery in Databases mit Übung** Knowledge discovery in databases with tutorial	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Richard Lenz
5	**Inhalt**	Theoretical knowledge on: • Why data mining? • What is data mining? • A multi-dimensional view of data mining • What kinds of data can be mined? • What kinds of patterns can be mined? • What technologies are used? • What kinds of applications are targeted? • Major issues in data mining • A brief history of data mining Practical exercises on: • Introduction to Pandas & scikit-learn • Data analysis & data preprocessing • Frequent Pattern • Classification • Clustering • Outlier
6	**Lernziele und Kompetenzen**	Die Studierenden • kennen den typischen KDD-Prozess; • kennen Verfahren zur Vorbereitung von Daten für das Data Mining; • definieren Distanz- oder Ähnlichkeits-Funktionen auf einem speziellen Datenbestand; • überprüfen Attribute eines Datensatzes auf ihre Bedeutung für die Analyse hin und transformieren ggf. Attributwerte geeignet; • wissen, wie ein typisches Data Warehouse aufgebaut ist; • kennen die Definition von Distanz- bzw. Ähnlichkeitsfunktionen für die verschiedenen Typen von Attributen; • sind vertraut mit dem Prinzip des Apriori-Algorithmus zur Bestimmung von Mengen häufiger Elemente (frequent itemsets); • kennen den FP-Growth-Algorithmus zum schnellen Auffinden von Mengen häufiger Elemente; • geben die Definitionen von Support und Confidence für Assoziationsregeln wieder; • beschreiben die Ermittlung von Assoziationsregeln auf der Basis von Mengen häufiger Elemente; • sind in der Lage, die Vorgehensweise bei Klassifikationsaufgaben darzustellen; • legen dar, wie ein Entscheidungsbaum auf einem Trainingsdatensatz erzeugt wird;

		• stellen das Prinzip der Bayes'schen Klassifikation dar; • zählen verschiedene Clustering-Verfahren auf; • beschreiben den Ablauf von k-Means-Clustering; • kennen die verschiedenen Arten von Ausreißern. • können die verschiedenen Schritte eines KDD Prozesses auch praktisch anwenden. The students: • know the typical KDD process; • know procedures for the preparation of data for data mining; • know the definition of distance or similarity functions for the different kinds of attributes; • define distance and similarity functions for a particular dataset; • check attributes of a dataset for their meaning with reference to an analysis and transform attribute values accordingly, if required. • know how a typical data warehouse is structured; • are familar with the principle of the Apriori algorithm for the identification of frequent itemsets; • know the FP-growth algorithm for a faster identification of frequent itemsets: • present the definitions of support and confidence for association rules; • describe the construction of association rules based on frequent itemsets; • are capable of describing the course of action in classification tasks; • present the construction of a decision tree based on a training dataset; • present the principle of Bayes' classification; • enumerate different clustering procedures; • describe the steps of k-means clustering; • know the different kinds of outliers. • are able to practically apply the various steps of a KDD process.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Datenbanksysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur mit MultipleChoice (90 Minuten)
11	**Berechnung der Modulnote**	Klausur mit MultipleChoice (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch

16	**Literaturhinweise**	The lecture is based on the following book:
		•
		  ° A. Géron, Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems, 2nd ed. OReilly Media, 2017, ISBN: 978-1491962299
		  ° H. Du, Data Mining Techniques and Applications: An Introduction. Cengage Learning EMEA, May 2010, p. 336, ISBN: 978-1844808915
		  ° I. H. Witten, E. Frank, M. A. Hall, et al., Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques, 4th. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016, ISBN: 0128042915

Vertiefungsrichtung Diskrete Simulation

1	**Modulbezeichnung** 97090	**Simulation und Modellierung I** Simulation and modelling I	**5 ECTS**
2	Lehrveranstaltungen	Übung: Exercises to Simulation and Modeling I (2 SWS) Vorlesung: Simulation and Modeling I (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Lisa Maile Jonathan Fellerer Prof. Dr. Reinhard German	

4	**Modulverantwortliche/r**	Prof. Dr. Reinhard German
5	**Inhalt**	Das Modul vermittelt die Grundlagen der diskreten Ereignissimulation und beinhaltet • diskrete Simulation • analytische Modellierung (z.B. Warteschlangen) • Eingabemodellierung (z.B. Fitting-Verfahren) • Zufallszahlenerzeugung • statistische Ausgabeanalyse • Modellierungsparadigmen (u.a. Ereignis-/Prozessorientierung, Warteschlangen, Automaten, Petri-Netze, UML, graphische Bausteine) • kontinuierliche und hybride Simulation • Simulationssoftware • Fallstudien Content: Overview of the various kinds of simulation • discrete simulation (computational concepts, simulation of queuing systems, simulation in Java, professional simulation tools) • required probability concepts and statistics, modeling paradigms (e.g., event/process oriented, queuing systems, Petri nets, UML statecharts) • input modeling (selecting input probability distributions) • random number generation (linear congruential generators and variants, generating random variates) • output analysis (warm-up period detection, independent replications, result presentation) • continuous and hybrid simulation (differential equations, numerical solution, hybrid statecharts) • simulation software, case studies, parallel and distributed simulation.
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben Kenntnisse über Verfahren und Realisierungsmöglichkeiten der diskreten Simulation mit Ausblick auf andere Simulationsarten • erwerben Kenntnisse über statistische Aspekte der Simulation, die für die Anwendung wichtig sind • wenden statistische Methoden zur Analyse und Bewertung von Eingabe- sowie Ausgabedaten an

		• erwerben praktische Erfahrung mit kommerziellen Simulationswerkzeugen • erwerben Erfahrungen bei der Simulation in verschiedenen Anwendungsbereichen (u.a. Rechnernetze, Fertigungssysteme, Materialflusssysteme) • entwickeln eigenständig anhand von Beispielaufgaben Simulationsmodelle unter Verwendung verschiedener Modellierungsparadigmen • können in Gruppen kooperativ und verantwortlich arbeiten Learning targets and competences: Students • gain knowledge about methods and realization possibilities of discrete simulation with an outlook on other types of simulation • gain knowledge of statistical aspects of simulation that are important for practice • apply statistical methods for analysis and evaluation of input and output data • gain hands-on experience with commercial simulation tools • gain experience in simulation in various fields of application (including computer networks, manufacturing systems, material flow systems) • independently develop simulation models on the basis of sample tasks using different modeling paradigms • can work in groups cooperatively and responsibly
7	**Voraussetzungen für die Teilnahme**	elementare Programmierkenntnisse, vorzugsweise in Java, Mathematikkenntnisse in Analysis, wie z.B. im 1. Semester der angewandten Mathematik vermittelt Recommended background knowledge: basic programming skills, preferably in Java, mathematics skills in analysis, such as taught in the first semester in applied mathematics.
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Diskrete Simulation Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) Prüfungsleistung/examination: Klausur, benotet, 5 ETCS/written exam, graded, 5 ETCS Dauer (in Minuten)/duration (in minutes): 90 Anteil an der Berechnung der Modulnote/Share in the calculation of the module grade: 100.0 % Die im Rahmen der Übung gestellten (zwei)wöchentlichen Übungsaufgaben müssen bestanden werden, um das Gesamtmodul anrechnen lassen zu können. Die Übung gilt als bestanden, wenn mindestens 50% der Punkte korrekt bearbeitet wurden. Die Bearbeitung erfolgt in Gruppen von 3 oder 4 Studenten. Die Abgabe erfolgt in Präsenz zu dedizierten Übungsterminen. Werden mindestens 75% der Punkte erreicht, werden der bestandenen schriftlichen Prüfung

		Bonuspunkte entsprechend einer Notenstufe (-0.3 oder -0.4 in der Endnote) hinzugefügt. ---------- The (bi-)weekly exercise tasks must be passed in order to receive credit for the entire module. The exercise is considered to be passed if at least 50% of the points have been correctly processed. The work is done in groups of 3 or 4 students. The submission is done in presence on dedicated exercise dates. If at least 75% of the points are achieved, bonus points corresponding to one grade level (-0.3 or -0.4 in the final grade) will be added to the passed written examination.
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Law, "Simulation Modeling and Analysis, 5th ed., McGraw Hill, 2014

Vertiefungsrichtung Elektronik und Informationstechnik

1	**Modulbezeichnung** 96401	**Globale Navigationssatellitensysteme** Global navigation satellite systems	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übung zu Globale Navigationssatellitensysteme (1 SWS) Vorlesung: Globale Navigationssatellitensysteme (3 SWS)	- 5 ECTS
3	Lehrende	Prof. Dr. Jörn Thielecke	

4	**Modulverantwortliche/r**	Prof. Dr. Jörn Thielecke
5	**Inhalt**	*Hinweis:* 1. Mehrere Übungsstunden werden rechnergestützt (Python) sein, um den Vorlesungsstoff durch eigene praktische Erfahrung zu vertiefen. 2. Eine Laborbesichtigung beim Fraunhofer Institut für Integrierte Schaltungen soll den Studierenden Einblick in die einschlägigen Arbeiten zu GPS und Galileo geben. *Inhalte:* *|1. Überblick: Signale und Systeme|* • Einführung • GPS Global Positioning System • Galileo • Satellitenergänzungssysteme: EGNOS, WAAS, LAAS • Mathematische Grundlagen: Navigationssignale, Gold Codes, Cramer-Rao-Schranke für Laufzeitmessungen *|2. Grundlagen und Funktiosweise der Satellitenortung|* • Koordinatensysteme, Zeitsysteme und Orbits • Ausbreitungsbedingungen und Fehlerquellen • Positions-, Geschwindigkeits- und Zeitschätzung • Hochgenaue Positionsschätzung mittels Trägerphase *|3. GNSS Empfänger|* • Signalkonditionierung • Leistungsfähigkeit der GPS- und Galileo-Signale • Releschleifen zur Signalverfolgung
6	**Lernziele und Kompetenzen**	1. Eine Übersicht über die Möglichkeiten von GPS und Galileo soll die Beurteilungsfähigkeit der Studierenden für neue Anwendungen schärfen. 2. Durch vertiefte Kenntnisse der Grundlagen, Funktionsweise und Fehlerquellen sollen die Studierenden die gelösten Herausforderungen und die Grenzen von GPS und Galileo einschätzen lernen. 3. Die Studierenden sollen ein nachrichtentechnisches Verständnis für die Funktionsweise eines GPS-Empfängers erlangen.
7	**Voraussetzungen für die Teilnahme**	Keine formalen Voraussetzungen, grundlegende Kenntnisse erforderlich in: linearer Algebra, Signal- & Systemtheorie, Wahrscheinlichkeitstheorie
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Elektronik und Informationstechnik Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	schriftlich oder mündlich (90 Minuten)
11	**Berechnung der Modulnote**	schriftlich oder mündlich (100%) Bei bestandener Prüfung wird die Note um eine Teilnotenstufe (z.B. von 2,0 auf 1,7) verbessert, wenn Sie: 1. Mindestens 75% der Hausaufgaben bestanden haben UND 2. Mindestens 75% der Rechnerübungen erfolgreich absolviert haben. Eine Note besser als 1,0 wird nicht vergeben.
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	1. Pratap Misra, Per Enge, Global Positioning System", Ganga-Jamuna Press, 2001 2. E.D. Kaplan, C.J. Hegarty, Understanding GPS Principles and Applications"Artech House, 2. Auflage, 2006 3. Werner Mansfeld, "Satellitenortung und Navigation, Vieweg, 2004

1	**Modulbezeichnung** 96801	**Kommunikationsstrukturen** Communication structures	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Kommunikationsstrukturen (2 SWS) Übung: Übungen zu Kommunikationsstrukturen (2 SWS)	5 ECTS 5 ECTS
3	Lehrende	Jürgen Frickel	

4	**Modulverantwortliche/r**	Jürgen Frickel
5	**Inhalt**	Einführung • Information und Kommunikation • Anwendungsgebiete - Kommunikation **Strukturen und Eigenschaften von Kommunikationssystemen** • Grundlegende Definitionen und Klassifikationen • Grundlegende Strukturen **Protokolle und Schnittstellen** • Grundlagen • Basis-Verfahren und Beispiele • TCP/IP-Protokol • Referenzmodell nach ISO/OSI • Sicherungsschicht/Data Link Layer (LLC und MAC) • Bitübertragungsschicht/Physical Layer • Übertragungsmedien **Hardware in Kommunikationsstrukturen** • HW-Architekturen und Funktionsblöcke • Digitale und Analoge Komponenten • Schaltungsdetails von Komponenten **Grundlagen von Bussystemen** • Klassifikation • Funktionale Eigenschaften • Arbitrierungs-Verfahren **Leitungsgebundene Anwendungen für Rechnersysteme** • Bus-Applikationen • Baustein-/IC-interne Busse (AMBA, FPI, ConTraBus, .) • Baugruppeninterne Busse (I2C, Chipsätze+Bridges, .) • Busse für Rechensysteme (VME, ISA, PCI, PCIe, AGP, .) • Peripherie-Busse (ATA, IEC, USB, Firewire, Fibre Channel, Thunderbolt .) **Leitungsgebundene Anwendungen in Systemen** • Feldkommunikation • Automobil, Luftfahrt, Space (CAN, MOST, LIN, MILBus, Spacewire .) • Industrie, Haustechnik (Profibus, EIB, .) • Weitverkehrsnetze • SDH, PDH, ATM,
6	**Lernziele und Kompetenzen**	1. Die Studierenden werden in die Lage versetzt die Konzepte und Verfahren vor allem drahtgebundener Kommunikationssysteme anzuwenden.

		2. Die Studierenden lernen die Funktionsweise und den Einsatzzweck diverser Kommunikationsprotokolle zu verstehen, und miteinander zu vergleichen. 3. Desweiteren analysieren und klassifizieren Sie grundlegende Strukturen von leitungsgebundenen Kommunikationssystemen anhand ihrer funktionalen Eigenschaften.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Elektronik und Informationstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	schriftlich oder mündlich (90 Minuten)
11	**Berechnung der Modulnote**	schriftlich oder mündlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 652213	**Satellitengestützte Ortsbestimmung** Global navigation satellite systems	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übung zu Globale Navigationssatellitensysteme (1 SWS) Vorlesung: Globale Navigationssatellitensysteme (3 SWS)	- 5 ECTS
3	Lehrende	Prof. Dr. Jörn Thielecke	

4	**Modulverantwortliche/r**	Prof. Dr. Jörn Thielecke
5	**Inhalt**	*Hinweis:* 1. Mehrere Übungsstunden werden rechnergestützt (MATLAB) sein, um den Vorlesungsstoff durch eigene praktische Erfahrung zu vertiefen. 2. Eine Laborbesichtigung beim Fraunhofer Institut für Integrierte Schaltungen soll Ihnen Einblick in die einschlägigen Arbeiten zu GPS und Galileo geben. *Inhalte:* *|1. Überblick: Signale und Systeme|* • Einführung • GPS Global Positioning System • Galileo • Satellitenergänzungssysteme: EGNOS, WAAS, LAAS • Mathematische Grundlagen: Navigationssignale, Gold Codes, Cramer-Rao-Schranke für Laufzeitmessungen *|2. Grundlagen und Funktiosweise der Satellitenortung|* • Koordinatensysteme, Zeitsysteme und Orbits • Ausbreitungsbedingungen und Fehlerquellen • Positions-, Geschwindigkeits- und Zeitschätzung • Hochgenaue Positionsschätzung mittels Trägerphase *|3. GNSS Empfänger|* • Signalkonditionierung • Leistungsfähigkeit der GPS- und Galileo-Signale • Releschleifen zur Signalverfolgung
6	**Lernziele und Kompetenzen**	1. Eine Übersicht über die Möglichkeiten von GPS und Galileo soll Ihre Beurteilungsfähigkeit für neue Anwendungen schärfen. 2. Durch vertiefte Kenntnisse der Grundlagen, Funktionsweise und Fehlerquellen sollen Sie die gelösten Herausforderungen und die Grenzen von GPS und Galileo einschätzen lernen. 3. Sie sollen ein nachrichtentechnisches Verständnis für die Funktionsweise eines GPS-Empfängers erlangen.
7	**Voraussetzungen für die Teilnahme**	Keine formalen Voraussetzungen, grundlegende Kenntnisse erforderlich in: linearer Algebra, Signal- & Systemtheorie, Wahrscheinlichkeitstheorie
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Elektronik und Informationstechnik Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	1. Pratap Misra, Per Enge, Global Positioning System", Ganga-Jamuna Press, 2001 2. E.D. Kaplan, C.J. Hegarty, Understanding GPS Principles and Applications"Artech House, 2. Auflage, 2006 3. Werner Mansfeld, "Satellitenortung und Navigation, Vieweg, 2004

1	**Modulbezeichnung** 43460	**Satellitenkommunikation** Satellite communication	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Albert Heuberger
5	**Inhalt**	Nach einem historischen Rückblick zur Entwicklung der Satellitenkommunikation werden die einzelnen Komponenten eines typischen Gesamtsystems (Boden- und Raumsegment) näher betrachtet. Hierzu zählt der prinzipielle Aufbau von Trägerraketen, von Satelliten (Satellitenplattformen, Subsysteme, Nutzlasten), die meist genutzten Umlaufbahnen und die verschiedenen Kommunikationsverbindungen (Uplink, Downlink, Inter-Satellite-Link). Die Besonderheiten der Signalausbreitung und -übertragung über große Entfernungen zwischen Bodenstationen und Satelliten werden erklärt und mit Beispielen ergänzt. Dabei wird insbesondere eingegangen auf verwendete Frequenzen, Signaldispersion und -dämpfung, atmosphärische Effekte sowie Störeinflüsse der Weltraumumgebung. Die Architektur transparenter und regenerativer Kommunikationseinheiten wird ausführlich an Beispielen kommerziell verfügbarer Transponder und Onboard-Prozessoren erklärt. Die Prinzipien moderner, standardisierter Verfahren zur Signalaufbereitung und Übertragung von Video-/Bild und Audiosignalen über Satellit (z.B. MPEG, H.264/265, DVB-S/-S2/-S2X) werden erläutert und diskutiert. Dies umfasst Verfahren zur Quellencodierung, Kanalcodierung und Modulation, Kanalzugriff und -diversität. Außerdem wird auf die im Orbit und im kommerziellen Einsatz befindlichen Kommunikationssatelliten und der damit verbundenen großen Dienstevielfalt eingegangen wie z.B. bei TV- und Breitbandversorgung sowie in Mobilkommunikationssystemen. Abschließend werden einige Herausforderungen und Forschungsansätze im Zusammenhang mit den neuen Megakonstellationen und Next Generation High Throughput Satellites (HTS) für zukünftige Satellitensysteme vorgestellt. Die in der Vorlesung behandelten physikalischen, elektro- und nachrichtentechnischen Zusammenhänge werden in den ergänzenden Übungen mit Rechenbeispielen vertieft. Gliederung der Vorlesung: **1. Einführung:** Überblick über die Hauptkomponenten, Satelliten, Anwendungen und Dienste, sowie Orbits, Aufgaben und Frequenzen der Satellitennetzwerke **2. Historie der Satellitenkommunikation:** Wichtige Meilensteine, Entwicklung in Europa und Deutschland **3. Orbits und Konstellationen:** Keplersche Gesetze, Beschreibung von Orbits, verwendete Umlaufbahnen, Bodenspuren, erreichbare Abdeckung

4. Trägersysteme:
Trägerraketen, Entwicklung, Anbietermarkt, Nutzlastfähigkeit, Startplätze, Startverlauf
5. Satellitenaufbau:
Auswahl aktueller Satellitenplattformen, Satellitenaufbau, Plattformkomponenten, Montageschritte und Tests
6. Satellitennutzlast (Payload):
Komponenten, Industrielle Beispiele, Aufbau und Aufgaben der Payload, Transponderarchitekturen, Antennen
7. Signalausbreitung und Leistungsbilanz:
Signalausbreitung, Freiraumverluste, Signaldämpfung, Rauschen, Signal-Rausch-Verhältnis, Linkbudget
8. Weltraumumgebung: Weltraumumgebungsbedingungen, Einflüsse auf den Satelliten und die Elektronik der Nutzlast
9. Quellencodierung:
Audio-, Bild- und Videokompression das Content des Satellitenfernsehens
10. Signalmodulation und Kanalcodierung:
Signalkonstellationen, Modulation und Codes zur Fehlerkorrektur
11. Diversitäts- und Zugriffsverfahren:
Medium Access, Duplextechniken, Multiplexmethoden, Diversitätstechniken
12. Moderne Satellitenkommunikationssysteme:
Rundfunksysteme wie Sirius XM Satellite Radio, zellulare Internetversorgung mittels Satellitenkommunikation
13. Neueste Themen aus Forschung und Entwicklung

SatKom auf StudOn: http://www.studon.uni-erlangen.de/crs117969.html

After a historical retrospective about the developments in satellite communication, the core components of a typical satellite system (ground- and space-segment) are introduced. The principles and architectures of rockets/ carriers, satellites (platform, subsystems, payload), used orbits, and the various communication links (uplink, downlink, inter-satellite-link) are shown. The special features and properties of signal transmission over such large distances are explained and stuffed with examples. In particular, more details are provided on the used frequencies, signal dispersion and attenuation, atmospheric effects as well as impairments due to space environment. The architecture of transparent and regenerative communication payloads are described in detail, accompanied by corresponding examples of commercially used transponders and onboard-processors and their technology.
The principles of modern standardized methods for signal transmission and preparation of video-/image- and audio-signals via satellite, e.g., MPEG, H.264/265, DVB-S/-S2/-S2X, are illustrated and discussed. This includes methods for efficient source coding, channel coding and modulation, channel access and diversity schemes.

		Furthermore, the currently available communication satellites in orbit and the related variety of commercial services are introduced like, e.g., TV- and broadband services as well as mobile communication services and systems. Based on that, a few challenges and perspectives for research and development for future satellite systems are highlighted with respect to the upcoming new mega constellations and next generation high throughput satellites (HTS). The physical, electro-technical and communications concepts and schemes shown in the lectures are complemented by tutorials with sample calculations. Table of contents: **1. Introduction:** Overview of main components, satellites, applications and services, orbits, tasks, frequencies, satellite networks **2. History of satellite communications:** Major milestones, development in Europe and Germany **3. Orbits and constellations:** Kepler's laws, description of orbits, orbits used, ground tracks, achievable coverage **4. Launcher systems:** Launch vehicles, providers, payload capabilities, launch sites, launch history **5. Satellite structure:** Selection of current satellite platforms, satellite structure, platform components, assembly steps and tests **6. Payload:** Components, structure and tasks of payload, transponder architecture, antennas **7. Signal propagation and link budget:** Signal propagation, free space losses, signal attenuation, noise, signal to noise ratio, link budget **8. Space environment:** Space environmental conditions, influences on the satellites and payload electronics **9. Source coding:** Audio, image and video compression - the satellite TV broadcasting content **10. Signal modulation and channel coding:** Signal constellations, modulation and error correction coding **11. Diversity and access schemes:** Medium access, duplex methods, multiplex methods, diversity techniques **12. Modern satellite communications systems:** Broadcasting systems like Sirius XM Satellite Radio, satellite cellular broadband communication **13. Latest topics in research and development**
6	**Lernziele und Kompetenzen**	• Die Studierenden bekommen einen guten Überblick über alle Aspekte der Satellitenkommunikation inklusive Historie. • Die Studierenden lernen die weltweit führenden oder in Europa ansässigen Firmen und Organisationen kennen, die in den Bereichen Satellitenbau und -betrieb, Satellitendienste bzw. -anwendungen, sowie Forschung und Entwicklung tätig sind. • Die Studierenden können die Herausforderungen der Weltraumumgebung sowie Vor- und Nachteile verschiedener Orbits einschätzen und wichtige Kenngröße berechnen • Die Studierenden lernen die Signalverarbeitungsschritte im Sender, Satelliten und Empfänger kennen - von der

		Audio/Video-Quelle über Link-Budget-Berechnungen bis zur Datensenke. • Die Studierenden lernen den Aufbau und wichtige Kenngrößen von Satelliten, Konstellationen und Launchern kennen und dabei verwendete Konzepte zu unterscheiden und zu klassifizieren bzgl. deren Vor- und Nachteilen.
7	**Voraussetzungen für die Teilnahme**	Keine formalen Voraussetzungen
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Elektronik und Informationstechnik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	schriftlich oder mündlich (90 Minuten)
11	**Berechnung der Modulnote**	schriftlich oder mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Skriptum zur Lehrveranstaltung

Vertiefungsrichtung Hardware-Software-Co-Design

1	**Modulbezeichnung** 636348	**Cyber-Physical Systems** Cyber-physical systems	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übung zu Cyber-Physical Systems (2 SWS) Vorlesung: Cyber-Physical Systems (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Dr.-Ing. Torsten Klie	

4	**Modulverantwortliche/r**	Dr.-Ing. Torsten Klie
5	**Inhalt**	Klassische Computersysteme zeichnen sich durch eine strikte Trennung von realer und virtueller Welt aus. Moderne Steuerungssysteme, die z.B. in modernen Fahrzeugen verbaut sind und die aus einer Vielzahl von Sensoren und Aktoren bestehen, entsprechen diesem Bild nur sehr eingeschränkt. Diese Systeme, oft "Cyber-Physical Systems (CPS)" genannt, erkennen ihre physische Umgebung, verarbeiten diese Informationen und können die physische Umwelt auch koordiniert beeinflussen. Hierzu ist eine starke Kopplung von physischem Anwendungsmodell und dem Computer-Steuerungsmodell nötig. Im Unterschied zu Eingebetteten Systemen bestehen CPS meist aus vielen vernetzten Komponenten, die sich selbständig untereinander koordinieren. Diese Vorlesung spannt den Bogen von kontrolltheoretischen Grundlagen über Selbstorganisiationsprinzipien bis hin zu visionären Anwendungen aus den Bereichen Verkehr und Medizintechnik. Ferner werden Entwurfsmethoden für Cyber-Physical Systems vorgestellt.
6	**Lernziele und Kompetenzen**	Die Studierenden erläutern, was Cyber-Physical Systems sind und auf welchen technologischen Grundlagen sie aufbauen, insbesondere in den Bereichen Regelungstechnik, Ablaufplanung, Kommunikation und Selbstorganisation bewerten CPS in verschiedenen Anwendungsgebieten stellen den Entwurfsprozess von CPS dar, insbesondere die Modellierung und die grundlegende Programmierung entdecken wesentliche Herausforderungen beim Entwurf, Ausbringung und Einsatz von CPS.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Andrea Bondavalli, Sara Bouchenak und Hermann Kopetz (Hrsg.) Cyber-Physical Systems of Systems: Foundations – A Conceptual Model and Some Derivations: The AMADEOS Legacy. Springer 2016. • Otto Föllinger Regelungstechnik. Hüthig 1992. • Hilmar Jaschek und Holger Voos Grundkurs der Regelungstechnik. Oldenbourg 2010. • Jörg Kahlert Crash-Kurs Regelungstechnik. VDE Verlag 2010. • Peter Marwedel Embedded Systems Design – Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things, 4. Auflage. Springer 2021 • André Platzner Logic Foundations of Cyber-physical Systems. Springer 2018. • Wolfgang Schneider Praktische Regelungstechnik. Vieweg +Teubner 2008. • Walid M. Taha, Abd-Ehamid M. Taha und Johan Thunberg Cyber-physical Systems – A Model-based Approach. Springer 2021.

1	**Modulbezeichnung** 773774	**Eingebettete Systeme (Vorlesung mit erweiterten Übungen)** Embedded systems	**7,5 ECTS**
2	Lehrveranstaltungen	Übung: UE-ES (2 SWS) Vorlesung: Eingebettete Systeme (2 SWS) Übung: Erweiterte Übungen zu Eingebettete Systeme (2 SWS)	2,5 ECTS 2,5 ECTS 2,5 ECTS
3	Lehrende	Patrick Plagwitz Khalil Esper Dominik Walter Prof. Dr.-Ing. Jürgen Teich PD Dr.Ing. Frank Hannig	

4	**Modulverantwortliche/r**	Joachim Falk Prof. Dr.-Ing. Jürgen Teich
5	**Inhalt**	Das Modul, Eingebettete Systeme mit erweiterter Übung, thematisiert den Entwurf und die Implementierung eingebetteter Systeme unter Einsatz formaler Methoden und rechnergestützter Entwurfsverfahren. Unter eingebetteten Systemen versteht man Rechensysteme, die auf einen Anwendungsbereich zugeschnitten (z.B. mobile Kommunikationsgeräte, Chipkartensysteme, Industriesteuerungen, Unterhaltungselektronik, Medizintechnik) und in einen technischen Kontext eingebunden sind. Das große Interesse am systematischen Entwurf von heterogenen eingebetteten Systemen ist verursacht durch die steigende Vielfalt und Komplexität von Anwendungen für eingebettete Systeme, die Notwendigkeit, Entwurfs- und Testkosten zu senken sowie durch Fortschritte in Schlüsseltechnologien (Mikroelektronik, formale Methoden). ***Content:*** *The focus of this module is the design and implementation of embedded systems using formal methods and computer-aided design techniques. Embedded systems are computing systems tailored for a particular application (e.g., mobile communication devices, smart card systems, industrial control, consumer electronics, medical technology) and integrated into a technical context. The keen interest in the systematic design of heterogeneous embedded systems is driven by the increasing diversity and complexity of embedded system applications, the need to reduce design and test costs, and advances in key technologies (microelectronics, formal methods).*
6	**Lernziele und Kompetenzen**	Fachkompetenz - Wissen • Die Studierenden setzen sich mit einem aktuellen Forschungsgebiet auseinander. The students deal with a current field of research. • In den erweiterten Übungen lernen die Studierenden aktuelle Entwurfswerkzeuge für die Architektursynthese (Hardware) und Softwaresynthese vor Ort an den Rechnerarbeitsplätzen des Lehrstuhls kennen. In the extended exercises, the

		students learn about current design tools for architecture synthesis (hardware) and software synthesis on-site at the chair's computer workstations. Fachkompetenz - Verstehen • Die Studierenden verstehen grundlegende Konzepte des Entwurfs eingebetteter Systeme. The students become familiar with the fundamental concepts of designing of embedded systems. Fachkompetenz - Anwenden • Die Studierenden wenden grundlegende Algorithmen an zur Analyse und Optimierung von Hardware-Architekturen und Echtzeit-Softwaresystemen. The students apply basic algorithms to analyze and optimize hardware architectures and real-time software systems. • Die Studierenden erfassen den Hardware/Software-Entwurf von Systemen mit harten Beschränkungen. The students understand the hardware/software design of hard-constrained systems. • Die Studierenden wenden aktuelle Entwurfswerkzeuge, die auf den Rechnerarbeitsplätzen des Lehrstuhls installiert sind, an, um damit die Aufgaben der erweiterten Übungen unter Anleitung zu lösen. The students apply current design tools installed on the chair's computer workstations to solve the tasks of the extended exercises with the help of instructions. Sozialkompetenz • Die Studierenden lernen aktuelle Entwurfswerkzeuge für die Architektursynthese (Hardware) und Softwaresynthese kennen bei der kooperativen Bearbeitung der erweiterten Übung in Gruppen. The students learn about current design tools for architecture synthesis (hardware) and software synthesis by processing the extended exercises in groups cooperatively.
7	**Voraussetzungen für die Teilnahme**	Die Auswahl dieses Moduls schließt die Auswahl der Module „Eingebettete Systeme“ und „Eingebettete Systeme (Vorlesung mit Übungen)“ aus.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio (90 Minuten)
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	## Empfohlenes Buch zur Begleitung und Vertiefung: • Teich J., Haubelt C.: "Digitale Hardware/Software-Systeme: Synthese und Optimierung", Springer-Verlag, 2007, ISBN: 978-3-540-46822-6 ## Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/eingebettete-systeme/

1	**Modulbezeichnung** 604896	**Eingebettete Systeme (Vorlesung mit Übungen)** Embedded systems	**5 ECTS**
2	Lehrveranstaltungen	Übung: UE-ES (2 SWS) Vorlesung: Eingebettete Systeme (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Khalil Esper Dominik Walter Patrick Plagwitz PD Dr.Ing. Frank Hannig Prof. Dr.-Ing. Jürgen Teich	

4	**Modulverantwortliche/r**	Joachim Falk Prof. Dr.-Ing. Jürgen Teich
5	**Inhalt**	Schwerpunkt des Moduls ist der Entwurf und die Implementierung eingebetteter Systeme unter Einsatz formaler Methoden und rechnergestützter Entwurfsverfahren. Unter eingebetteten Systemen versteht man Rechensysteme, die auf einen Anwendungsbereich zugeschnitten (z.B. mobile Kommunikationsgeräte, Chipkartensysteme, Industriesteuerungen, Unterhaltungselektronik, Medizintechnik) und in einen technischen Kontext eingebunden sind. Das große Interesse am systematischen Entwurf von heterogenen eingebetteten Systemen ist verursacht durch die steigende Vielfalt und Komplexität von Anwendungen für eingebettete Systeme, die Notwendigkeit, Entwurfs- und Testkosten zu senken sowie durch Fortschritte in Schlüsseltechnologien (Mikroelektronik, formale Methoden). ***Content:*** *The focus of this module is the design and implementation of embedded systems using formal methods and computer-aided design techniques. Embedded systems are computing systems tailored for a particular application (e.g., mobile communication devices, smart card systems, industrial control, consumer electronics, medical technology) and integrated into a technical context. The keen interest in the systematic design of heterogeneous embedded systems is driven by the increasing diversity and complexity of embedded system applications, the need to reduce design and test costs, and advances in key technologies (microelectronics, formal methods).*
6	**Lernziele und Kompetenzen**	Fachkompetenz - Wissen • Die Studierenden setzen sich mit einem aktuellen Forschungsgebiet auseinander. The students deal with a current field of research. Fachkompetenz - Verstehen • Die Studierenden verstehen grundlegende Konzepte des Entwurfs eingebetteter Systeme. The students become familiar with the fundamental concepts of designing of embedded systems. Fachkompetenz - Anwenden

		• Die Studierenden wenden grundlegende Algorithmen an zur Analyse und Optimierung von Hardware-Architekturen und Echtzeit-Softwaresystemen. The students apply basic algorithms to analyze and optimize hardware architectures and real-time software systems. • Die Studierenden erfassen den Hardware/Software-Entwurf von Systemen mit harten Beschränkungen. The students understand the hardware/software design of hard-constrained systems.
7	**Voraussetzungen für die Teilnahme**	Die Auswahl dieses Moduls schließt die Auswahl der Module „Eingebettete Systeme (Vorlesung mit erweiterten Übungen)“ und „Eingebettete Systeme“ aus.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio (90 Minuten)
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	Empfohlenes Buch zur Begleitung und Vertiefung: • Teich J., Haubelt C.: "Digitale Hardware/Software-Systeme: Synthese und Optimierung", Springer-Verlag, 2007, ISBN: 978-3-540-46822-6 Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/eingebettete-systeme/

1	**Modulbezeichnung** 502509	**Hardware-Software-Co-Design** Hardware-software-co-design	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Joachim Falk Prof. Dr.-Ing. Jürgen Teich
5	**Inhalt**	Zahlreiche Realisierungen eingebetteter Systeme (z.B. Mobiltelephone, Faxgeräte, Industriesteuerungen) zeichnen sich durch kooperierende Hardware- und Softwarekomponenten aus. Die Popularität solcher Realisierungsformen lässt sich begründen durch 1) die steigende Vielfalt und Komplexität heterogener Systeme, 2) die Notwendigkeit, Entwurfs- und Testkosten zu senken und 3) Fortschritte in Schlüsseltechnologien (Mikroelektronik, formale Entwurfsmethoden). Zum Beispiel bieten Halbleiterhersteller kostengünstige ASICs an, die einen Mikrocontroller und benutzerspezifische Peripherie und Datenpfade auf einem Chip integrieren. Die Synthese solcher Systeme wirft jedoch eine Reihe neuartiger Entwurfsprobleme auf, insbesondere 1) die Frage der Auswahl von Hardware- und Softwarekomponenten, 2) die Partitionierung einer Spezifikation in Hard- und Software, 3) die automatische Synthese von Interface- und Kommunikationsstrukturen und 4) die Verifikation und Cosimulation. 1) Überblick und Vergleich von Architekturen und Komponenten in Hardware/Software-Systemen. 2) Aufbau eines Compilers und Codeoptimierungsverfahren für Hardware und Software 3) Hardware/Software-Partitionierung (Partitionierung komplexer Systeme, Schätzungsverfahren, Performanzanalyse, Codegenerierung) 4) Interfacesynthese (Kommunikationsarten, Synchronisation, Synthese) 5) Verifikation und Cosimulation 6) Tafelübungen Content: Numerous realisations of embedded systems (e.g. mobile phones, fax machines, industrial controls) are characterised by cooperating hardware and software components. The popularity of such realisations can be explained by 1) the increasing diversity and complexity of heterogeneous systems, 2) the need to reduce design and testing costs, and 3) advances in key technologies (microelectronics, formal design methods). For example, semiconductor manufacturers offer low-cost ASICs that integrate a microcontroller and user-specific peripherals and data paths on a single chip. However, the synthesis of such systems raises a number of novel design problems, in particular 1) the issue of hardware and software component selection, 2) the partitioning of a specification into hardware and software, 3) the automatic synthesis of interface and communication structures, and 4) verification and cosimulation.

		1) Overview and comparison of architectures and components in hardware/software systems. 2) Structure of a compiler and code optimisation procedures for hardware and software. 3) Hardware/software partitioning (partitioning of complex systems, estimation procedures, performance analysis, code generation) 4) Interface synthesis (communication types, synchronisation, synthesis) 5) Verification and cosimulation 6) Blackboard exercises
6	**Lernziele und Kompetenzen**	Fachkompetenz - Wissen • Die Studierenden erhalten Einblick in ein aktuelles Forschungsgebiet. Fachkompetenz - Verstehen • Die Studierenden verstehen Grundlagen des modernen Systementwurfs. • Die Studierenden erklären Implementierungsalternativen für digitale Hardware/Software-Systeme. Fachkompetenz - Anwenden • Die Studierenden wenden grundlegende Algorithmen an, zur Analyse und Optimierung von Hardware/Software-Systemen. Competneces: Professional competence - Knowledge • Students gain insight into a current field of research. Professional competence - Understanding • Students understand the basics of modern system design. • Students explain implementation alternatives for digital hardware/software systems. Professional competence - Application • Students apply basic algorithms to analyse and optimise hardware/software systems.
7	**Voraussetzungen für die Teilnahme**	Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Hardware-Software-Co-Design (Vorlesung mit erweiterter Übung)“ aus.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio (90 Minuten)
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	## Empfohlene Bücher zur Begleitung und Vertiefung: • Teich J., Haubelt C.: "Digitale Hardware/Software-Systeme: Synthese und Optimierung", Springer-Verlag, 2007, ISBN: 978-3-540-46822-6 • Gajski, D. et al.: "Specification and Design of Embedded Systems", Prentice Hall, Englewood Cliffs, NJ, 1994, ISBN: 978-0131507319 ## Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/hardware-software-co-design

1	**Modulbezeichnung** 292952	**Hardware-Software-Co-Design (Vorlesung mit erweiterter Übung)** Hardware-software-co-design (Lecture with extended exercises)	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Joachim Falk Prof. Dr.-Ing. Jürgen Teich
5	**Inhalt**	Zahlreiche Realisierungen eingebetteter Systeme (z.B. Mobiltelephone, Faxgeräte, Industriesteuerungen) zeichnen sich durch kooperierende Hardware- und Softwarekomponenten aus. Die Popularität solcher Realisierungsformen lässt sich begründen durch 1) die steigende Vielfalt und Komplexität heterogener Systeme, 2) die Notwendigkeit, Entwurfs- und Testkosten zu senken und 3) Fortschritte in Schlüsseltechnologien (Mikroelektronik, formale Entwurfsmethoden). Zum Beispiel bieten Halbleiterhersteller kostengünstige ASICs an, die einen Mikrocontroller und benutzerspezifische Peripherie und Datenpfade auf einem Chip integrieren. Die Synthese solcher Systeme wirft jedoch eine Reihe neuartiger Entwurfsprobleme auf, insbesondere 1) die Frage der Auswahl von Hardware- und Softwarekomponenten, 2) die Partitionierung einer Spezifikation in Hard- und Software, 3) die automatische Synthese von Interface- und Kommunikationsstrukturen und 4) die Verifikation und Cosimulation. 1) Überblick und Vergleich von Architekturen und Komponenten in Hardware/Software-Systemen. 2) Aufbau eines Compilers und Codeoptimierungsverfahren für Hardware und Software 3) Hardware/Software-Partitionierung (Partitionierung komplexer Systeme, Schätzungsverfahren, Performanzanalyse, Codegenerierung) 4) Interfacesynthese (Kommunikationsarten, Synchronisation, Synthese) 5) Verifikation und Cosimulation 6) Tafelübungen 7) Demonstrationen mit rechnergestützten Entwurfswerkzeugen und praktische Übungen Content: Numerous realisations of embedded systems (e.g. mobile phones, fax machines, industrial controls) are characterised by cooperating hardware and software components. The popularity of such realisations can be explained by 1) the increasing diversity and complexity of heterogeneous systems, 2) the need to reduce design and testing costs, and 3) advances in key technologies (microelectronics, formal design methods). For example, semiconductor manufacturers offer low-cost ASICs that integrate a microcontroller and user-specific peripherals and data paths on a single chip.

		However, the synthesis of such systems raises a number of novel design problems, in particular 1) the issue of hardware and software component selection, 2) the partitioning of a specification into hardware and software, 3) the automatic synthesis of interface and communication structures, and 4) verification and cosimulation. 1) Overview and comparison of architectures and components in hardware/software systems. 2) Structure of a compiler and code optimisation procedures for hardware and software. 3) Hardware/software partitioning (partitioning of complex systems, estimation procedures, performance analysis, code generation) 4) Interface synthesis (communication types, synchronisation, synthesis) 5) Verification and cosimulation 6) Blackboard exercises 7) Demonstrations with computer-aided design tools and practical exercises
6	**Lernziele und Kompetenzen**	Fachkompetenz - Wissen • Die Studierenden erhalten Einblick in ein aktuelles Forschungsgebiet. Fachkompetenz - Verstehen • Die Studierenden verstehen Grundlagen des modernen Systementwurfs. • Die Studierenden erklären Implementierungsalternativen für digitale Hardware/Software-Systeme. Fachkompetenz - Anwenden • Die Studierenden wenden grundlegende Algorithmen an, zur Analyse und Optimierung von Hardware/Software-Systemen. • Die Studierenden wenden das erlernte Wissen in den erweiterten Übungen vor Ort an den Rechnerarbeitsplätzen des Lehrstuhls an. Sozialkompetenz • Die Studierenden benutzen aktuelle Entwurfswerkzeuge für die Spezifikation, Optimierung und Prototypisierung von Hardware/Software-Systemen bei der kooperativen Bearbeitung der erweiterten Übung in Gruppen. Competences: Professional competence - Knowledge • Students gain insight into a current field of research. Professional competence - Understanding • Students understand the basics of modern system design. • Students explain implementation alternatives for digital hardware/software systems. Professional competence - Application • Students apply basic algorithms to analyse and optimise hardware/software systems. • The students apply the knowledge they have acquired in the extended exercises on site at the computer workstations of the department. Social competence

		• The students use current design tools for the specification, optimisation and prototyping of hardware/software systems in the cooperative processing of the extended exercise in groups.
7	**Voraussetzungen für die Teilnahme**	Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Hardware-Software-Co-Design“ aus.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio (90 Minuten) Klausur (90 min) und erfolgreicher Bearbeitung aller Übungsaufgaben in den erweiterten Übungen (verpflichtend, vor Ort an den Rechnerarbeitsplätzen des Lehrstuhls). Die Sprache der Klausur ist abhängig von der Wahl der Studierenden entweder Deutsch oder Englisch.
11	**Berechnung der Modulnote**	Portfolio (100%) Die Modulnote ergibt sich aus der Klausurnote.
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Empfohlene Bücher zur Begleitung und Vertiefung: • Teich J., Haubelt C.: "Digitale Hardware/Software-Systeme: Synthese und Optimierung", Springer-Verlag, 2007, ISBN: 978-3-540-46822-6 • Gajski, D. et al.: "Specification and Design of Embedded Systems", Prentice Hall, Englewood Cliffs, NJ, 1994, ISBN: 978-0131507319 Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/hardware-software-co-design

1	**Modulbezeichnung** 43510	**Parallele Systeme** Parallel systems	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Joachim Falk PD Dr.Ing. Frank Hannig Prof. Dr.-Ing. Jürgen Teich
5	**Inhalt**	Aktuelle PCs verfügen über Mehrkernprozessoren und Grafikkarten, die wiederum aus hunderten von einfachen Prozessoren bestehen können. Hierdurch wird ein hohes Maß an nebenläufiger Datenverarbeitung möglich, welche bis vor einigen Jahren nur in Großrechnern erreicht werden konnte. Die effiziente Ausnutzung dieser Parallelität bedarf allerdings mehr als nur mehrerer Prozessoren, insbesondere muss das zu lösende Problem Parallelverarbeitung erlauben. In dieser Vorlesung werden Eigenschaften unterschiedlicher paralleler Rechnerarchitekturen und Metriken zu deren Beurteilung behandelt. Weiterhin werden Modelle und Sprachen zum Programmieren paralleler Rechner eingeführt. Neben der Programmierung von allgemeinen Parallelrechnern werden Entwurfsmethoden (CAD) vorgestellt, wie man ausgehend von einer algorithmischen Problemstellung ein massiv paralleles Rechenfeld in VLSI herleiten kann, das genau dieses Problem optimal parallel berechnet. Solche Schaltungen spielen auf der Bit- bzw. Wortebene eine dominante Rolle (Arithmetik) sowie bei Problemen der Signal- und Bildverarbeitung (z.B. Filter). Schwerpunkt der Vorlesung ist die Vermittlung von Grundlagen der parallelen Datenverarbeitung. Im Einzelnen werden behandelt: 1) Theorie der Parallelität (parallele Computermodelle, parallele Spezifikationsformen und -sprachen, Performanzmodelle und -berechnung) 2) Klassifikation paralleler und skalierbarer Rechnerarchitekturen (Multiprozessoren und Multicomputer, Vektorrechner, Datenflussmaschinen, VLSI-Rechenfelder) 3) Programmierbare System-on-Chip (SoC) und Mehrkern-Architekturen (Grafik-Prozessoren, Cell, etc.) 4) Programmierung paralleler Rechner (Sprachen und Modelle, Entwurfsmethoden und Compiler, Optimierung) 5) Massive Parallelität: Vom Algorithmus zur Schaltung *Today's PCs consist of multi-core processors and graphics cards that again comprise hundreds to thousands of simple processors. As a result of this, a very high degree of parallel data processing becomes possible, which was subjected to supercomputers a couple of years ago. The efficient exploitation of parallel processing requires not only multiple processors but also parallelism inherent in the problem to be solved. In this lecture, properties of different parallel computer architectures and corresponding quality metrics are examined. Further, models and parallel programming languages are introduced. In addition*

		to programming general parallel computers, design methods (CAD) are presented that systematically transform an algorithmic problem description into a massive parallel processor array (VLSI), which can optimally execute the given problem in parallel. Such highly parallel circuits play an essential role at the bit level and circuit level (arithmetics) as well as in the case of signal processing and image processing (e.g., filter). The focus of this lecture are foundations of parallel data processing. *In detail, the following topics are covered:* 1) Theory of parallelism (parallel models of computation, parallel specification and parallel languages, performance models) 2) Classification of parallel and scalable computer architectures (multi-processors and multi-computers, vector computers, data-flow machines, VLSI processor arrays) 3) Programmable System-on-Chip (SoC) and multi-core architetcures (graphics processors, Cell, etc.) 4) Programming of parallel computers (languages and models, design methods and compiler, optimization) 5) Massive parallelism: From algorithm to circuit
6	**Lernziele und Kompetenzen**	Schwerpunkt der Vorlesung ist die Vermittlung von Grundlagen der parallelen Datenverarbeitung. *The focus of this lecture are foundations of parallel data processing.* Fachkompetenz - Verstehen • Die Studierenden verstehen grundlegende Konzepte der parallelen Datenverarbeitung, sowohl theoretischer Art anhand von Modellen, als auch an Architekturbeispielen. The students become familiar with the fundamentals of parallel data processing, theoretic in the form of models as well as by architecture examples. Fachkompetenz - Anwenden • Die Studierenden setzen sich mit modernen eingebetteten parallelen Ein-Chip-Architekturen auseinander. The students get familiar with modern embedded parallel system-on-chip architectures. • Die Studierenden wenden grundlegende Performanzmodelle und Parallelisierungtechniken zur Analyse und Optimierung von parallelen Algorithmen und Architekturen an. The students exercise basic performance models and parallelization techniques for the analysis and optimization of parallel algorithms and architectures. • Die Studierenden setzen die Modellierung und den Entwurf von massiv-parallelen Prozessorfeldern in konkreten Aufgaben selbstständig um. In concrete tasks, the students apply independently the modeling and the design of massively parallel processors arrays.
7	**Voraussetzungen für die Teilnahme**	Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Parallele Systeme (Vorlesung mit erweiterten Übungen)“ aus.

8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio (90 Minuten) • Ein Wechsel der Prüfungsform von einer Klausur zu einer mündlichen Prüfung ist in Ausnahmefällen (siehe § 16 ABMPO/TechFak) auch nach Semesterbeginn noch möglich. In diesem Fall werden die Studierenden spätestens zwei Wochen nach Vorlesungsbeginn informiert. • Prüfungssprache: Deutsch oder Englisch. Die Unterrichts- und Prüfungssprache hängt von den Sprachkenntnissen und Präferenzen der Teilnehmerinnen und Teilnehmer ab und wird dementsprechend innerhalb der ersten zwei Wochen nach Vorlesungsbeginn festgelegt.
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/parallele-systeme

1	**Modulbezeichnung** 740665	**Parallele Systeme (Vorlesung mit erweiterten Übungen)** Parallel systems with extended exercises	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Joachim Falk PD Dr.Ing. Frank Hannig Prof. Dr.-Ing. Jürgen Teich
5	**Inhalt**	Aktuelle PCs verfügen über Mehrkernprozessoren und Grafikkarten, die wiederum aus hunderten von einfachen Prozessoren bestehen können. Hierdurch wird ein hohes Maß an nebenläufiger Datenverarbeitung möglich, welche bis vor einigen Jahren nur in Großrechnern erreicht werden konnte. Die effiziente Ausnutzung dieser Parallelität bedarf allerdings mehr als nur mehrerer Prozessoren, insbesondere muss das zu lösende Problem Parallelverarbeitung erlauben. In dieser Vorlesung werden Eigenschaften unterschiedlicher paralleler Rechnerarchitekturen und Metriken zu deren Beurteilung behandelt. Weiterhin werden Modelle und Sprachen zum Programmieren paralleler Rechner eingeführt. Neben der Programmierung von allgemeinen Parallelrechnern werden Entwurfsmethoden (CAD) vorgestellt, wie man ausgehend von einer algorithmischen Problemstellung ein massiv paralleles Rechenfeld in VLSI herleiten kann, das genau dieses Problem optimal parallel berechnet. Solche Schaltungen spielen auf der Bit- bzw. Wortebene eine dominante Rolle (Arithmetik) sowie bei Problemen der Signal- und Bildverarbeitung (z.B. Filter). Im Einzelnen werden behandelt: 1) Theorie der Parallelität (parallele Computermodelle, parallele Spezifikationsformen und -sprachen, Performanzmodelle und -berechnung) 2) Klassifikation paralleler und skalierbarer Rechnerarchitekturen (Multiprozessoren und Multicomputer, Vektorrechner, Datenflussmaschinen, VLSI-Rechenfelder) 3) Programmierbare System-on-Chip (SoC) und Mehrkern-Architekturen (Grafik-Prozessoren, Cell, etc.) 4) Programmierung paralleler Rechner (Sprachen und Modelle, Entwurfsmethoden und Compiler, Optimierung) 5) Massive Parallelität: Vom Algorithmus zur Schaltung 6) Praktische Übungen mit rechnergestützten Werkzeugen *Today's PCs consist of multi-core processors and graphics cards that again comprise hundreds to thousands of simple processors. As a result of this, a very high degree of parallel data processing becomes possible, which was subjected to supercomputers a couple of years ago. The efficient exploitation of parallel processing requires not only multiple processors but also parallelism inherent in the problem to be solved. In this lecture, properties of different parallel computer architectures and corresponding quality metrics are examined. Further,*

		models and parallel programming languages are introduced. In addition to programming general parallel computers, design methods (CAD) are presented that systematically transform an algorithmic problem description into a massive parallel processor array (VLSI), which can optimally execute the given problem in parallel. Such highly parallel circuits play an essential role at the bit level and circuit level (arithmetics) as well as in the case of signal processing and image processing (e.g., filter). *In detail, the following topics are covered:* 1) Theory of parallelism (parallel models of computation, parallel specification and parallel languages, performance models) 2) Classification of parallel and scalable computer architectures (multi-processors and multi-computers, vector computers, data-flow machines, VLSI processor arrays) 3) Programmable System-on-Chip (SoC) and multi-core architetcures (graphics processors, Cell, etc.) 4) Programming of parallel computers (languages and models, design methods and compiler, optimization) 5) Massive parallelism: From algorithm to circuit 6) Practical training with computer-aided design tools
6	**Lernziele und Kompetenzen**	Schwerpunkt der Vorlesung ist die Vermittlung von Grundlagen der parallelen Datenverarbeitung. *The focus of this lecture are foundations of parallel data processing.* Fachkompetenz - Verstehen • Die Studierenden verstehen grundlegende Konzepte der parallelen Datenverarbeitung, sowohl theoretischer Art anhand von Modellen, als auch an Architekturbeispielen. The students become familiar with the fundamentals of parallel data processing, theoretic in the form of models as well as by architecture examples. Fachkompetenz - Anwenden • Die Studierenden setzen sich mit modernen eingebetteten parallelen Ein-Chip-Architekturen auseinander. The students get familiar with modern embedded parallel system-on-chip architectures. • Die Studierenden wenden grundlegende Performanzmodelle und Parallelisierungtechniken zur Analyse und Optimierung von parallelen Algorithmen und Architekturen an. The students exercise basic performance models and parallelization techniques for the analysis and optimization of parallel algorithms and architectures. • Die Studierenden setzen die Modellierung und den Entwurf von massiv-parallelen Prozessorfeldern in konkreten Aufgaben selbstständig um. In concrete tasks, the students apply independently the modeling and the design of massively parallel processors arrays. • Die Studierenden wenden das erlernte Wissen in den erweiterten Übungen vor Ort an den Rechnerarbeitsplätzen

		des Lehrstuhls an. The students apply their learned knowledge in hands-on computer exercises on-site at the chair's computer workstations.
7	**Voraussetzungen für die Teilnahme**	Die Auswahl dieses Moduls schließt die Auswahl des Moduls „Parallele Systeme“ aus.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio (90 Minuten) • Prüfung und erfolgreiche Bearbeitung aller Übungsaufgaben in den erweiterten Übungen (verpflichtend, vor Ort an den Rechnerarbeitsplätzen des Lehrstuhls). • Ein Wechsel der Prüfungsform von einer Klausur zu einer mündlichen Prüfung ist in Ausnahmefällen (siehe § 16 ABMPO/TechFak) auch nach Semesterbeginn noch möglich. In diesem Fall werden die Studierenden spätestens zwei Wochen nach Vorlesungsbeginn informiert. • Prüfungssprache: Deutsch oder Englisch. Die Unterrichts- und Prüfungssprache hängt von den Sprachkenntnissen und Präferenzen der Teilnehmerinnen und Teilnehmer ab und wird dementsprechend innerhalb der ersten zwei Wochen nach Vorlesungsbeginn festgelegt.
11	**Berechnung der Modulnote**	Portfolio (100%) Die Modulnote ergibt sich aus der Prüfungsleistung.
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/parallele-systeme

1	**Modulbezeichnung** 43190	**Reconfigurable Computing** Reconfigurable computing	**7,5 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Joachim Falk
5	**Inhalt**	**Content:** Reconfigurable (adaptive) computing is a novel yet important research field investigating the capability of hardware to adapt to changing computational requirements such as emerging standards, late design changes, and even to changing processing requirements arising at run-time. Reconfigurable computing thus benefits from a) the programmability of software similar to the Von Neumann computer and b) the speed and efficiency of parallel hardware execution. The purpose of the course reconfigurable computing is to instruct students about the possibilities and rapidly growing interest in adaptive hardware and corresponding design techniques by providing them the necessary knowledge for understanding and designing reconfigurable hardware systems and studying applications benefiting from dynamic hardware reconfiguration. After a general introduction about benefits and application ranges of reconfigurable (adaptive) computing in contrast to general-purpose and application-specific computing, the following topics will be covered: • Reconfigurable computing systems: Introduction of available technology including fine grained look up table (LUT-) based reconfigurable systems such as field programmable gate arrays (FPGA) as well as newest coarse grained architectures and technology. • Design and implementation: Algorithms and steps (design entry, functional simulation, logic synthesis, technology mapping, place and route, bit stream generation) to implement (map) algorithms to FPGAs. The main focus lies on logic synthesis algorithms for FPGAs, in particular LUT technology mapping. • Temporal partitioning: techniques to reconfigure systems over time. Covered are the problems of mapping large circuits which do not fit one single device. Several temporal partitioning techniques are studied and compared. • Temporal placement: Techniques and algorithms to exploit the possibility of partial and dynamic (run-time) hardware reconfiguration. Here, OS-like services are needed that optimize the allocation and scheduling of modules at run-time. • On-line communication: Modules dynamically placed at run-time on a given device need to communicate as well as transport data off-chip. State-of-the-art techniques are introduced how modules can communicate data at run-time

		including bus-oriented as well as network-on-a-chip (NoC) approaches. • Designing reconfigurable applications on Xilinx Virtex FPGAs: In this part, the generation of partial bitstreams for components to be placed at run-time on Xilinx FPGAs is introduced and discussed including newest available tool flows. • Applications: This section presents applications benefiting from dynamic hardware reconfiguration. It covers the use of reconfigurable systems including rapid prototyping, reconfigurable supercomputers, reconfigurable massively parallel computers and studies important application domains such as distributed arithmetic, signal processing, network packet processing, control design, and cryptography.
6	**Lernziele und Kompetenzen**	**Learning objectives and competencies:** Domain-specific knowledge • The students know to exploit run-time reconfigurable design methodologies for adaptive applications. Domain-specific comprehension • The students understand the mapping steps, and optimization algorithms. • The students classify different types and kinds of reconfigurable hardware technologies available today. • The students clarify pros and cons of reconfigurable computing technology. • The students summarize applications benefiting from reconfigurable computing. Domain-specific practice • The students apply design tools for implementation of circuits and systems-on-a-chip (SoC) on FPGAs during practical training. Social competency • The students perform group work in small teams during practical training.
7	**Voraussetzungen für die Teilnahme**	Selection of this module prohibits the selection of the modules "Reconfigurable Computing (Lecture with Exercises)" or "Reconfigurable Computing (Lecture with Extended Exercises)" by the student.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio Oral examination (Duration: 30 min) and successful completion of all tasks of the extended exercises (mandatory, at the workstations residing in our lab at the chair).
11	**Berechnung der Modulnote**	Portfolio (100%) The oral examination determines the final grade of the module.
12	**Turnus des Angebots**	nur im Wintersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	**Further reading material:** • The Hamburg VHDL Archive (see Documentation link for free books) http://tams-www.informatik.uni-hamburg.de/research/vlsi/vhdl/index.php • Interactive VHDL Tutorial with 150 examples from ALDEC http://www.aldec.com/downloads/ • Easy FPGA tutorials, projects, and boards http://www.fpga4fun.com • Xilinx WebPack ISE and Modelsim MXE (free FPGA synthesis tool and free VHDL simulator) http://www.xilinx.com/ise/logic_design_prod/webpack.htm • Symphony EDA free VHDL simulator (select FREE Edition license) http://www.symphonyeda.com/products.htm • Icarus open-source Verilog simulator http://www.icarus.com/eda/verilog/ **Further information:** https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/reconfigurable-computing/

1	**Modulbezeichnung** 741941	**Reconfigurable Computing (Lecture with Exercises)** Reconfigurable computing (lecture with exercises)	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Reconfigurable Computing (2 SWS) Übung: Exercises to Reconfigurable Computing (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr.-Ing. Jürgen Teich Tobias Hahn Pierre-Louis Sixdenier	

4	**Modulverantwortliche/r**	Joachim Falk Prof. Dr.-Ing. Jürgen Teich
5	**Inhalt**	**Content:** Reconfigurable (adaptive) computing is a novel yet important research field investigating the capability of hardware to adapt to changing computational requirements such as emerging standards, late design changes, and even to changing processing requirements arising at run-time. Reconfigurable computing thus benefits from a) the programmability of software similar to the Von Neumann computer and b) the speed and efficiency of parallel hardware execution. The purpose of the course reconfigurable computing is to instruct students about the possibilities and rapidly growing interest in adaptive hardware and corresponding design techniques by providing them the necessary knowledge for understanding and designing reconfigurable hardware systems and studying applications benefiting from dynamic hardware reconfiguration. After a general introduction about benefits and application ranges of reconfigurable (adaptive) computing in contrast to general-purpose and application-specific computing, the following topics will be covered: • Reconfigurable computing systems: Introduction of available technology including fine grained look up table (LUT-) based reconfigurable systems such as field programmable gate arrays (FPGA) as well as newest coarse grained architectures and technology. • Design and implementation: Algorithms and steps (design entry, functional simulation, logic synthesis, technology mapping, place and route, bit stream generation) to implement (map) algorithms to FPGAs. The main focus lies on logic synthesis algorithms for FPGAs, in particular LUT technology mapping. • Temporal partitioning: techniques to reconfigure systems over time. Covered are the problems of mapping large circuits which do not fit one single device. Several temporal partitioning techniques are studied and compared. • Temporal placement: Techniques and algorithms to exploit the possibility of partial and dynamic (run-time) hardware reconfiguration. Here, OS-like services are needed that optimize the allocation and scheduling of modules at run-time.

		• On-line communication: Modules dynamically placed at run-time on a given device need to communicate as well as transport data off-chip. State-of-the-art techniques are introduced how modules can communicate data at run-time including bus-oriented as well as network-on-a-chip (NoC) approaches. • Designing reconfigurable applications on Xilinx Virtex FPGAs: In this part, the generation of partial bitstreams for components to be placed at run-time on Xilinx FPGAs is introduced and discussed including newest available tool flows. • Applications: This section presents applications benefiting from dynamic hardware reconfiguration. It covers the use of reconfigurable systems including rapid prototyping, reconfigurable supercomputers, reconfigurable massively parallel computers and studies important application domains such as distributed arithmetic, signal processing, network packet processing, control design, and cryptography.
6	**Lernziele und Kompetenzen**	**Learning objectives and competencies:** Domain-specific knowledge • The students know to exploit run-time reconfigurable design methodologies for adaptive applications. Domain-specific comprehension • The students understand the mapping steps and optimization algorithms. • The students classify different types and kinds of reconfigurable hardware technologies available today. • The students clarify pros and cons of reconfigurable computing technology. • The students summarize applications benefiting from reconfigurable computing. • The students describe the design of circuits and systems-on-a-chip (SoC) on FPGAs.
7	**Voraussetzungen für die Teilnahme**	Selection of this module prohibits the selection of the modules "Reconfigurable Computing" or "Reconfigurable Computing (Lecture with Extended Exercises)" by the student.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Further reading material: • The Hamburg VHDL Archive (see Documentation link for free books) http://tams-www.informatik.uni-hamburg.de/research/vlsi/vhdl/index.php • Interactive VHDL Tutorial with 150 examples from ALDEC http://www.aldec.com/downloads/ • Easy FPGA tutorials, projects, and boards http://www.fpga4fun.com • Xilinx WebPack ISE and Modelsim MXE (free FPGA synthesis tool and free VHDL simulator) http://www.xilinx.com/ise/logic_design_prod/webpack.htm • Symphony EDA free VHDL simulator (select FREE Edition license) http://www.symphonyeda.com/products.htm • Icarus open-source Verilog simulator http://www.icarus.com/eda/verilog/ Further information: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/reconfigurable-computing/

1	**Modulbezeichnung** 110334	**Reconfigurable Computing (Lecture with Extended Exercises)** Reconfigurable computing (lecture with extended exercises)	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Reconfigurable Computing (2 SWS) Übung: Extended Exercises to Reconfigurable Computing (2 SWS) Übung: Exercises to Reconfigurable Computing (2 SWS)	2,5 ECTS 2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr.-Ing. Jürgen Teich Marcel Brand Pierre-Louis Sixdenier Tobias Hahn	

4	**Modulverantwortliche/r**	Joachim Falk Prof. Dr.-Ing. Jürgen Teich
5	**Inhalt**	**Content:** Reconfigurable (adaptive) computing is a novel yet important research field investigating the capability of hardware to adapt to changing computational requirements such as emerging standards, late design changes, and even to changing processing requirements arising at run-time. Reconfigurable computing thus benefits from a) the programmability of software similar to the Von Neumann computer and b) the speed and efficiency of parallel hardware execution. The purpose of the course reconfigurable computing is to instruct students about the possibilities and rapidly growing interest in adaptive hardware and corresponding design techniques by providing them the necessary knowledge for understanding and designing reconfigurable hardware systems and studying applications benefiting from dynamic hardware reconfiguration. After a general introduction about benefits and application ranges of reconfigurable (adaptive) computing in contrast to general-purpose and application-specific computing, the following topics will be covered: • Reconfigurable computing systems: Introduction of available technology including fine grained look up table (LUT-) based reconfigurable systems such as field programmable gate arrays (FPGA) as well as newest coarse grained architectures and technology. • Design and implementation: Algorithms and steps (design entry, functional simulation, logic synthesis, technology mapping, place and route, bit stream generation) to implement (map) algorithms to FPGAs. The main focus lies on logic synthesis algorithms for FPGAs, in particular LUT technology mapping. • Temporal partitioning: techniques to reconfigure systems over time. Covered are the problems of mapping large circuits which do not fit one single device. Several temporal partitioning techniques are studied and compared.

		• Temporal placement: Techniques and algorithms to exploit the possibility of partial and dynamic (run-time) hardware reconfiguration. Here, OS-like services are needed that optimize the allocation and scheduling of modules at run-time. • On-line communication: Modules dynamically placed at run-time on a given device need to communicate as well as transport data off-chip. State-of-the-art techniques are introduced how modules can communicate data at run-time including bus-oriented as well as network-on-a-chip (NoC) approaches. • Designing reconfigurable applications on Xilinx Virtex FPGAs: In this part, the generation of partial bitstreams for components to be placed at run-time on Xilinx FPGAs is introduced and discussed including newest available tool flows. • Applications: This section presents applications benefiting from dynamic hardware reconfiguration. It covers the use of reconfigurable systems including rapid prototyping, reconfigurable supercomputers, reconfigurable massively parallel computers and studies important application domains such as distributed arithmetic, signal processing, network packet processing, control design, and cryptography.
6	**Lernziele und Kompetenzen**	**Learning objectives and competencies:** Domain-specific knowledge • The students know to exploit run-time reconfigurable design methodologies for adaptive applications. Domain-specific comprehension • The students understand the mapping steps, and optimization algorithms. • The students classify different types and kinds of reconfigurable hardware technologies available today. • The students clarify pros and cons of reconfigurable computing technology. • The students summarize applications benefiting from reconfigurable computing. Domain-specific practice • The students apply design tools for implementation of circuits and systems-on-a-chip (SoC) on FPGAs during practical training. Social competency • The students perform group work in small teams during practical training.
7	**Voraussetzungen für die Teilnahme**	Selection of this module prohibits the selection of the modules "Reconfigurable Computing" or "Reconfigurable Computing (Lecture with Exercises)" by the student.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	Portfolio Oral examination (Duration: 30 min) and successful completion of all tasks of the extended exercises (mandatory, at the workstations residing in our lab at the chair).
11	**Berechnung der Modulnote**	Portfolio (100%) The oral examination determines the final grade of the module.
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Further reading material: • The Hamburg VHDL Archive (see Documentation link for free books) http://tams-www.informatik.uni-hamburg.de/research/vlsi/vhdl/index.php • Interactive VHDL Tutorial with 150 examples from ALDEC http://www.aldec.com/downloads/ • Easy FPGA tutorials, projects, and boards http://www.fpga4fun.com • Xilinx WebPack ISE and Modelsim MXE (free FPGA synthesis tool and free VHDL simulator) http://www.xilinx.com/ise/logic_design_prod/webpack.htm • Symphony EDA free VHDL simulator (select FREE Edition license) http://www.symphonyeda.com/products.htm • Icarus open-source Verilog simulator http://www.icarus.com/eda/verilog/ Further information: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/reconfigurable-computing/

1	**Modulbezeichnung** 92280	**Verifikation digitaler Systeme**	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übung zur Verifikation digitaler Systeme (2 SWS) Vorlesung: Verifikation digitaler Systeme (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr. Oliver Keszöcze	

4	**Modulverantwortliche/r**	Prof. Dr. Oliver Keszöcze
5	**Inhalt**	Für den Entwurf eines digitalen Systems werden heute in der Industrie ebenso viele Verifikationsingenieure wie Designer benötigt. Trotzdem beansprucht die Verifikation heute bereits 70%-80% der gesamten Entwurfszeit. Neben konventionellen Verifikationserfahren wie der Simulation sind werden seit einigen Jahren sogenannte "formale Verifikationsmethoden" in heutigen Entwursflüssen eingesetzt. Der Umgang mit diesen Methoden stellt ein wichtiges neues Aufgabenfeld dar. Im Gegensatz zur Simulation beruht die formale Verifikation auf exakten mathematischen Methoden zum Nachweis funktionaler Schaltungseigenschaften. Dadurch können Entwurfsfehler frühzeitiger und mit höherer Zuverlässigkeit als bisher erkannt werden. Jedes System zur formalen Hardwareverifikation erfordert: 1. ein geeignetes Modell des zu verifizierenden Systems 2. eine Sprache zur Formulierung der zu verifizierenden Eigenschaften 3. eine Beweismethode. Die Vorlesung behandelt diese drei Bereiche, vermittelt die grundlegenden Algorithmen und Konzepte moderner Werkzeuge für die formale Hardwareverifikation und erläutert deren Einsatz in der industriellen Praxis. Im Einzelnen werden in dieser Vorlesung die folgenden Punkte behandelt: 1. Modellierung digitaler Systeme 2. Unterschiede formaler und simulationsbasierter Verifikationsmethoden 3. Äquivalenzvergleich 4. Formale und simulationsbasierte Eigenschaftsprüfung 5. Assertions 6. Verifikation arithmetischer Schaltungen
6	**Lernziele und Kompetenzen**	Fachkompetenz -- Verstehen • Die Studierenden erläutern die wesentlichen Techniken zur Verifikation digitaler Systeme, zeigen deren Vor- und Nachteile auf und vergleichen diese bezüglich Ihrer Mächtigkeit und Komplexität. Fachkompetenz -- Anwenden • Die Studierenden wenden Modellierungstechniken aus den Bereichen der Binären Entscheidungsdiagramme inkl. bekannter Erweiterungen (MDD, etc.) sowie der SAT-Löser auf Systembeschreibungen an. • Die Studierenden Verifikationstechniken aus den Bereichen der formalen Äquivalenz- und Eigenschaftsprüfung (Model

		Checking, Symbolic Model Checking, Bounded Model Checking) auf Systembeschreibungen an.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Vertiefungsrichtung Kommunikationssysteme

1	**Modulbezeichnung** 43950	**Kommunikationssysteme** Communication systems	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Communication Systems (2 SWS) Übung: Übungen zu Kommunikationssysteme (ÜKS) (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr. Reinhard German Mamdouh Muhammad	

4	**Modulverantwortliche/r**	Prof. Dr. Reinhard German
5	**Inhalt**	Aus Rechnerkommunikation ist der grundlegende Aufbau von IP-basierten Netzen bekannt, Inhalt von Kommunikationssysteme sind weitere Netztechnologien wie Leitungsvermittlung (Telefonnetze, Sonet/SDH/WDM) und Netze mit virtueller Leitungsvermittlung (ATM, MPLS) sowie Netzwerkvirtualisierung (SDN, NFV), Multimediakommunikation über paketvermittelte Netze (Streaming, RTP, SIP, Multicast), Dienstgüte in paketvermittelten Netzen (Integrated Services, RSVP, Differentiated Services, Active Queue Management, Policing, Scheduling), drahtlose und mobile Kommunikation (GSM, UMTS, LTE, 5G, Wimax, WLAN, Bluetooth, ZigBee u.a. Sensornetze). Auch Kommunikation in der Industrie wird behandelt. In der Übung werden praktische Aufgaben im Labor durchgeführt: ein Labor enthält mehrere IP-Router, Switches und Rechner, IP-Telefone und Telefonie-Software für VoIP, es werden verschiedene Konfigurationen eingestellt und getestet. Ein weiterer Übungsteil beschäftigt sich mit Mobilkommunikation. *Contents:* Based on the course computer communications the architecture of IP networks is known. Contents of this course will be additional networking technologies such as circuit switching (telephony, SONET/SDH/WDM) and networks with virtual circuit switching (ATM, MPLS) as well as network virtualization (SDN, NFV), multimedia communications over packet switched networks (streaming, RTP, SIP, multicast), quality-of-service in packet switched networks (integrated services, RSVP, differentiated services, active queue management, policing, scheduling), wireless and mobile communications (GSM, UMTS, LTE, 5G, Wimax, WLAN, Bluetooth, sensor networks such as ZigBee). Industrial communication will also be a topic. In the tutorial practical tasks are performed in the laboratory: One laboratory contains several IP routers, switches and computers, IP phones and VoIP telephone software. Various configurations are set up and tested. Another part of the tutorial deals with mobile communications.
6	**Lernziele und Kompetenzen**	Die Studierenden erlangen Kenntnisse über Technologien bei der Leitungs- und Paketvermittlung in leitungsgebundenen und drahtlosen/mobilen Netzen Kenntnisse über die Grundlagen von Dienstgütemechanismen in paketvermittelten Netzen praktische Erfahrung in der Konfiguration eines IP-Switch-Router-Netzes mit Multimediaverkehr Students obtain the following learning targets and competences

		Knowledge of technologies in circuit and packet switching in wired and wireless/mobile networks Knowledge of the foundations of quality of service mechanisms in packet switched networks Practical experience in configuring an IP switch router network with multimedia traffic
7	**Voraussetzungen für die Teilnahme**	Basic knowledge of working with the Linux command line interface (terminal).
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Kommunikationssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	schriftlich oder mündlich (90 Minuten) Prüfungsleistung, Klausur, Dauer (in Minuten): 90, benotet, 5 ECTS Anteil an der Berechnung der Modulnote: 100.0 % Die Modulprüfung besteht aus: • Bearbeitung (zwei)wöchentlicher Aufgabenblätter in Gruppenarbeit. Für diese unbenotete Studienleistung sind alle Aufgabenblätter korrekt zu lösen und abzugeben. • Klausur von 90 Minuten
11	**Berechnung der Modulnote**	schriftlich oder mündlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Kurose, Ross. Computer Networking: A Top-Down Approach. 7th Ed., Pearson Education, 2017 W. Stallings. Data and Computer Communications, 10th ed., Pearson Education, 2014 W. Stallings. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud, Pear-son Education, 2016 Cox. An Introduction to LTE. Wiley, 2012

1	**Modulbezeichnung** 858896	**Modellierung, Optimierung und Simulation von Energiesystemen** Modeling, optimization and simulation of energy systems	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Marco Pruckner
5	**Inhalt**	In der Vorlesung Modellierung, Optimierung und Simulation von Energiesystemen werden systemtechnische Planungs- und Analysemethoden behandelt, die zur Lösung komplexer und interdisziplinärer Entscheidungsaufgaben in der Energiewirtschaft eingesetzt werden. Dabei werden die wichtigsten Methoden und Verfahren anhand praktischer Fragestellungen (z.B. Ausbau erneuerbarer Energien, Zunahme der Elektromobilität) aus der energiepolitischen Planung vermittelt und die Bewältigung technisch-ökonomischer Probleme verdeutlicht. Zu den eingesetzten Tools zählen die Statistiksoftware R, AnyLogic und lpSolve. Vorkenntnisse im Umgang mit diesen Werkzeugen ist nicht zwingend erforderlich. In den Übungen werden Einführungen in die genannten Softwarepakete gegeben.
6	**Lernziele und Kompetenzen**	Die Studierenden • unterscheiden Probleme und Herausforderungen, die mit dem Energieumstieg verbunden sind, • erfassen die Vorteile und die Anwendungsmöglichkeiten computergestützter Planungsmethoden im Energiebereich, • analysieren verschiedene Problemstellungen und setzen Lösungen dafür um, • erlernen verschiedene Methoden der Datenanalyse, Optimierung und Simulation.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Kommunikationssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Deutsch oder Englisch
16	**Literaturhinweise**	

Vertiefungsrichtung Künstliche Intelligenz

1	**Modulbezeichnung** 894856	**Künstliche Intelligenz I** Artificial intelligence I	**7,5 ECTS**
2	Lehrveranstaltungen	Übung: KI I - Ü (2 SWS) Vorlesung: Artificial Intelligence I (4 SWS)	- 7,5 ECTS
3	Lehrende	PD Dr. Florian Rabe Prof. Dr. Michael Kohlhase	

4	**Modulverantwortliche/r**	Prof. Dr. Michael Kohlhase
5	**Inhalt**	Dieses Modul beschäftigt sich mit den Grundlagen der Künstlichen Intelligenz (KI), insbesondere formale Wissensrepräsentation, Heuristische Suche, Automatisches Planen und Schliessen unter Unsicherheit. --- This module covers the foundations of Artificial Intelligence (AI), in particular symbolic techniques based on search and inference.
6	**Lernziele und Kompetenzen**	- Wissen: Die Studierenden lernen grundlegende Repräsentationsformalismen und Algorithmen der Künstlichen Intelligenz kennen. - Anwenden: Die Konzepte werden an Beispielen aus der realen Welt angewandt (Übungsaufgaben). - Analyse: Die Studierenden lernen die über die modellierung in der Maschine menschliche Intelligenzleistungen besser einzuschätzen. Sozialkompetenz - Die Studierenden arbeiten in Kleingruppen zusammen um kleine Projekte zu bewältigen *Inhalt*: - Agentenmodelle als Grundlage der Künstlichen Intelligenz - Logisches Programmieren in Prolog - Heuristische Suche als Methode zur Problemlösung - Zwei-Agenten-Suche (automatisierung von Brettspielen) mittels heuristischer Suche - Constraint Solving/Propagation - Logische Sprachen für die Wissensrepräsentation - Inferenz and Automatisiertes Theorembeweisen (DPLL-Varianten und PL1)_ - Classisches Planen - Planen und Agieren in der wirklichen Welt. --- Technical, Learning, and Method Competencies - Knowledge: The students learn foundational representations and algorithms in AI. - Application: The concepts learned are applied to examples from the real world (homeworks). - Analysis: By modeling human cognitive abilities, students learn to assess and understand human intelligence better. - Social Competences: Students work in small groups to solve an AI game-play challenge/competition (Kalah). Contents: Foundations of symbolic AI, in particular:

		- Agent Models as foundation of AI - Logic Programming in Prolog - Heuristic Search as a methdod for problem solving - Adversarial Search (automating board games) via heuristic search - Constraint Solving/Propagation - Logical Languages for knowledge representation - Inference and automated theorem proving - Classical Planning - Planning and Acting in the real world.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Künstliche Intelligenz Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio (90 Minuten)
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Die Vorlesung folgt weitgehend dem Buch Stuart Russell und Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition, 2009. Deutsche Ausgabe: Stuart Russell und Peter Norvig: Künstliche Intelligenz: Ein Moderner Ansatz. Pearson-Studium, 2004 (Übersetzung der 2. Auflage). ISBN: 978-3-8273-7089-1.

1	**Modulbezeichnung** 532733	**Künstliche Intelligenz II** Artificial intelligence II	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Michael Kohlhase
5	**Inhalt**	Dieses Modul beschäftigt sich mit den Grundlagen der Künstlichen Intelligenz (KI), insbesondere mit Techniken des Schließens unter Unsicherheit, des maschinellen Lernens und der Sprachverarbeitung. Das Modul baut auf dem Modul Künstliche Intelligenz I vom Wintersemester auf und führt dieses weiter.
6	**Lernziele und Kompetenzen**	Fach- Lern- bzw. Methodenkompetenz - Wissen: Die Studierenden lernen grundlegende Repräsentationsformalismen und Algorithmen der Künstlichen Intelligenz kennen. - Anwenden: Die Konzepte werden an Beispielen aus der realen Welt angewandt (Übungsaufgaben). - Analyse: Die Studierenden lernen über die Modellierung in der Maschine menschliche Intelligenzleistungen besser einzuschätzen. Sozialkompetenz - Die Studierenden arbeiten in Kleingruppen zusammen um kleine Projekte zu bewältigen. Inhalte: • Inferenz unter Unsicherheit • Bayessche Netzwerke • Rationale Entscheidungstheorie (MDPs and POMDPs) • Machinnelles Learnend und Neuronale Netzwerke • Verarbeitung Natürlicher Sprache --- This course covers the foundations of Artificial Intelligence (AI), in particular reasoning under uncertainty, machine learning and (if there is time) natural language understanding. This course builds on the course Artificial Intelligence I from the preceding winter semester and continues it. Learning Goals and Competencies Technical, Learning, and Method Competencies • Knowledge: The students learn foundational representations and algorithms in AI. • Application: The concepts learned are applied to examples from the real world (homeworks). • Analysis: By modeling human cognitive abilities, students learn to assess and understand human intelligence better. • Social Competences: Students work in small groups to solve the and machine learning challenge/competition. Contents: • Inference under Uncertainty

		• Bayesian Networks • Rational Decision Theory (MDPs and POMDPs) • Machine Learning and Neural Networks • Natural Language Processing
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Künstliche Intelligenz Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio (90 Minuten)
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Die Vorlesung folgt weitgehend dem Buch Stuart Russell und Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition, 2009. Deutsche Ausgabe: Stuart Russell und Peter Norvig: Künstliche Intelligenz: Ein Moderner Ansatz. Pearson-Studium, 2004 (Übersetzung der 2. Auflage). ISBN: 978-3-8273-7089-1. Literature The course follows the following textbook: Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition, 2009.

1	**Modulbezeichnung** 675137	**Logik-Basierte Sprachverarbeitung** Logic-based speech representation	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Logik-Basierte Sprachverarbeitung (4 SWS)	5 ECTS
3	Lehrende	Prof. Dr. Michael Kohlhase	

4	**Modulverantwortliche/r**	Prof. Dr. Michael Kohlhase
5	**Inhalt**	Grundlagen der logikbasierten Sprachverarbeitung, Syntax, Semantik-Konstruktion, und Semantische Verarbeitung natürlicher Sprache im Grammatical Framework (GF)
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden beherrschen moderne, sehr expressive Formalismen zur Syntaktisch/Semantishen Analyse und Bedeutungsrepräsentation natürlicher Sprache. Sie können eingeschränkt neue Formalismen entwickeln. Anwenden Die Studierenden entwickeln Grammatiken und Bedeutungsrepräsentationen im Meta-Framework GF und setzen diese durch Interprätationsabbildungen in Verbindung. Analysieren Die Studierenden analysieren die innere Struktur natürlichsprachlicher Phrasen und Sätze. Sie wählen für eine zu repräsentierenden Phänomene geeignete Formalismen aus. Lern- bzw. Methodenkompetenz Die Studierenden erarbeiten selbständig formale syntaktische und semantische Repräsentationen für Natürliche Sprache Sozialkompetenz Die Studierenden arbeiten in Kleingruppen erfolgreich zusammen. Inhalt Grundlagen der logikbasierten Sprachverarbeitung, Syntax, Semantik-Konstruktion, und Semantisch/Pragmatischen Verarbeitung natürlicher Sprache im Grammatical/Logical/Inferential Framework (GLIF) Insbesondere - Montague's "Method of Fragments" - Grammatikentwicklung in GF - Logikentwicklung und Domänenrepräsentation in MMT - Representation von Inferenzsystemen in LF - Inferenzbasierte Modelleriung der Semantisch/proagmatischen Verarbeitung
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5

9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Künstliche Intelligenz Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 221236	**Logik-basierte Wissensrepräsentation für mathematisch/technisches Wissen** Logic-based knowldege representation for mathematic/ technical knowledge	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Michael Kohlhase
5	**Inhalt**	Grundlagen der Mathematik, Modulare Formalisierung in Theoriegraphen, Narrative Strukturen in informellen mathematisch/ technischen Dokumenten, Formalisierung von Logiksprachen in Metalogiken.
6	**Lernziele und Kompetenzen**	Fachkompetenz • Wissen: Die Studierenden beherrschen moderne, sehr expressive Formalismen (Syntax, Semantik, Beweistheorie) zur Repräsentation komplexen Wissens. Sie können diese selbst in Metalogiken repräsentieren und eigeschränkt neue Formalismen entwickeln. • Anwenden:Die Studierenden formalisieren komplexe mathematisch/technische Sachverhalte in jeweils geeigneten Meta-Sprachen und setzen diese durch Interprätationsabbildungen in Verbindung. • Analysieren: Die Studierenden analysieren die innere Struktur komplexer Objekte, Konzepte, und Modelle. Sie wählen für eine zu repräsentierende mathematisch/technische Domäne geeignete Formalismen aus. • Lern- bzw. Methodenkompetenz: Die Studierenden erarbeiten selbständig formale Repräsentationen und Beweise. • Sozialkompetenz: Die Studierenden arbeiten in Kleingruppen erfolgreich zusammen. **Inhalt** • Grundlagen der Mathematik • Modulare Formalisierung in Theoriegraphen, • Narrative Strukturen in informellen mathematisch/technischen Dokumenten, • Formalisierung von Logiksprachen in Metalogiken.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Künstliche Intelligenz Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio

11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93134	**Wissensrepräsentation und -verarbeitung** Knowledge representation and processing	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Wissensrepräsentation und -verarbeitung (4 SWS) Übung: ÜWuV (2 SWS)	7,5 ECTS -
3	Lehrende	Prof. Dr. Michael Kohlhase PD Dr. Florian Rabe	

4	**Modulverantwortliche/r**	Prof. Dr. Michael Kohlhase PD Dr. Florian Rabe
5	**Inhalt**	Dieses Modul führt allgemein und grundlegend in die Wissensrepräsentation und -Verarbeitung ein. Dies beinhaltet alle Aspekte von Wissensrepräsentationssprachen und Wissen wie zum Beispiel Ontologiesprachen und Linked Data, Programmiersprachen und Algorithmen, Datenbeschreibungssprachen und Daten-Mengen, Logik und Beweise sowie natürliche Sprache und informelle Dokumente. Die Vorlesung behandelt all diese Aspekte grundlegend und vergleichend und geht eingehend auf die Integration und Interoperabilität der verschiedenen Aspekte ein. Die Übung vertieft dies im praktischen Umgang mit state-of-the-art Software-Systemen für die verschiedenen Aspekte. Das Modul kann belegt werden sowohl als Einstieg in weitere Module im Rahmen der Vertiefungsrichtung Künstlichen Intelligenz im Bachelor oder Master als auch als einmalige Überblicksvorlesung.
6	**Lernziele und Kompetenzen**	Die Teilnehmer erlernen und verstehen grundlegende Konzepte der Wissensrepräsentation und wie sie sie in der Praxis anwenden können. Konkret erlernen sie Wissensrepräsentationssprachen aus dem Bereich Ontologiesprachen, Programmiersprachen, Datenbeschreibungssprachen, Logik sowie natürliche Sprache. Sie verstehen die jeweiligen Vor- und Nachteile der und die Interrelationen zwischen den Sprachen. Sie erlernen, wie sie zu gegebenen Wissensrepräsentations-Problemen passende Sprachen auswählen, einsetzen und kombinieren können.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Künstliche Intelligenz Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	

Vertiefungsrichtung Mustererkennung

1	**Modulbezeichnung** 23070	**Biomedizinische Signalanalyse** Biomedical signal analysis	**5 ECTS**
2	Lehrveranstaltungen	Übung: BioSig-UE (2 SWS) Vorlesung: Biomedizinische Signalanalyse (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr. Björn Eskofier Daniel Krauß Katharina Jäger	

4	**Modulverantwortliche/r**	Prof. Dr. Björn Eskofier Daniel Krauß
5	**Inhalt**	**Inhalt** Im Rahmen der Vorlesung werden (a) die Grundlagen der Generation von wichtigen Biosignalen im menschlichen Körper, (b) die Messung von Biosignalen und (c) Methoden zur Analyse von Biosignalen erläutert und dargestellt. Behandelte Biosignale sind unter anderem Aktionspotential (AP), Elektrokardiogramm (EKG), Elektromyogramm (EMG), Elektroenzephalogramm (EEG), oder Mechanomyogramm (MMG). Bei der Messung liegt der Fokus beispielsweise auf der Messtechnik oder der korrekten Sensor- bzw. Elektrodenanbringung. Im größten Teil der Vorlesung, Analyse von Biosignalen, werden Konzepte zur Filterung für die Artefaktreduktion, der Wavelet Analyse, der Ereigniserkennung und der Wellenformanalyse behandelt. Zum Schluss wird einen Einblick in überwachte Verfahren der Mustererkennung gegeben. Für weitere Informationen, besuchen Sie bitte unseren zugehörigen StudOn Kurs. **Content** The lecture content explains and outlines (a) basics for the generation of important biosignals of the human body, (b) measurement of biosignals, and (c) methods for biosignals analysis. Considered biosignals are among others action potential (AP), electrocardiogram (ECG), electromyogram (EMG), electroencephalogram (EEG), or mechanomyogram (MMG). The focus during the measurement part is for example the measurement technology or the correct sensor and electrode placement. The main part of the lecture is the analysis part. In this part, concepts like filtering for artifact reduction, wavelet analysis, event detection or waveform analysis are covered. In the end, an insight into pattern recognition methods is obtained. For more information, please visit our associated StudOn course
6	**Lernziele und Kompetenzen**	**Die Studierenden können nach erfolgreichem Abschluss des Kurses** **Fachkompetenz** **Wissen** • die Entstehung, Messung und Charakteristika der wichtigsten Biosignale des menschlichen Körpers wiedergeben **Verstehen**

- die wesentlichen Ursachen von Artefakten in Biosignalen erklären
- Zusammenhänge zwischen der Entstehung der Biosignale des menschlichen Körper und dem gemessenen Signal erklären
- Messmethoden der wichtigsten Biosignale erklären
- Filteroperationen zur Eliminierung von Artefakten erläutern
- bekannte Algorithmen der Verarbeitung bestimmter Biosignal erklären (z.B. Pan Tompkins für EKG)
- typische Komponenten und ihre Bedeutung in einer generischen Signalanalyse Kette erläutern
- die Struktur und Funktionsweise von Systemen zur maschinellen Klassifikation einfacher Muster darstellen

Anwenden

- Signalcharakteristiken im Zeit- und Frequenzbereich bestimmen
- Algorithmen der Biosignalverarbeitung anwenden und in Python implementieren
- Filteroperationen zur Eliminierung von Artefakten anwenden und in Python implementieren
- Methoden selbstständig auf interdisziplinäre Fragestellungen der Medizin und der Ingenieurwissenschaften anwenden
- das Ergebnis von typischen Filteroperationen abschätzen

Analysieren

- Filtercharakteristika von Schaltkreisen ableiten
- Algorithmen der Biosignalverarbeitung vergleichen
- Klassifikationsprobleme in Python lösen
- Typische Artefakte in Biosignalen erkennen und Lösungsstrategien vorschlagen

Evaluieren (Beurteilen)

- Biosignale mit medizinischen Normalwerten vergleichen und im medizinischen Kontext evaluieren
- Klassifikationsergebnisse beurteilen
- die Bedeutung der Biosignalverarbeitung für die Medizintechnik diskutieren
- Probleme in Gruppen kooperativ und verantwortlich lösen und in der Übungsgruppe bzw. im Forum diskutieren

After completion of the course, students are able to

Knowledge

- reproduce the generation, measurement, and characteristics of important biosignals of the human body

Understanding

- explain the causes of artifacts in biosignals
- explain relations between the generation of biosignals and the measured signal
- explain methods for the measurement of important biosignals
- explain filter operations for the reduction of artifacts

		• explain algorithms for the analysis of important biosignals (e.g. Pan Tompkins for EKG) • explain typical components and their importance in the signal analysis chain • explain the structure and functioning of systems for machine learning and pattern recognition **Application** • determine signal characteristics in the time and frequency domain • apply and implement algorithms for signal analysis in Python • implement filter operations for the reduction of artifacts in Python • estimate the result of filter operations • apply methods to interdisciplinary problems in medicine and medical engineering **Analyze** • derive filter characteristics from electric circuits • compare signal analysis algorithms • solve classification problems in Python • recognize typical artifacts in biosignals and propose solutions for their reduction **Evaluation** • compare biosignals with medical norm values and evaluate them in a medical context • evaluate classification results • discuss the importance of biomedical signal analysis for medical engineering • solve and discuss problems in groups cooperatively in the group excercises and the online forum

7	**Voraussetzungen für die Teilnahme**	**Prerequisites**
		The Biosig lectures and exercises do not have formal requirements. However, we expect you to have some knowledge about the following topics:
		• Basics of Physiology and Anatomy (High-school level)
		•
		  ° Basic elements of electronic circuits (resistor, capacitor, inductor) and related equations
		• Basic math: Integration, Differentiation, Limits
		•
		  ° Fourier Transform (qualitative understanding)
		  ° Basic filter types
		  ° z-plane (qualitative understanding)
		
		Furthermore, some knowledge in the following topics will be beneficial to easily understand the content of the lecture:
		
		• Advanced filter concepts
		• z-plane math / z-transform / pole-zero plots

		• Frequency domain math / detailed understanding of Fourier transform and its properties • Laplace transform • Basics of Python (for the exercises) If you want to refresh your knowledge on all the aforementioned topics, we recommend the following lectures and online resources: Note that some of them go beyond the requirements of this lecture for many topics! • Signals and Systems I • Grundlagen der Anatomie und Physiologie für Medizintechniker • Video Series: Introduction to discrete Control (and further videos from this channel, as general introduction to filter and z-plane math) • A visual introduction to Fourier Transform • Udacity Python Course Course materials from the Stanford "Introduction to Scientific Python"
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Nebenfach Artificial Intelligence in Biomedical Engineering Bachelor of Science Informatik 20222 Nebenfach Medizinische Informatik Bachelor of Science Informatik 20222 Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222 Electronic Exam (in presence), 90min.
10	**Studien- und Prüfungsleistungen**	elektronische Prüfung
11	**Berechnung der Modulnote**	elektronische Prüfung (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• R.M. Rangayyan, Biomedical Signal Analysis: A case-study approach. 1st ed., 2002, New York, NY: John Wiley & Sons. • E.N. Bruce, Biomedical Signal Processing and Signal Modeling. 1st ed., 2001, New York, NY: John Wiley & Sons.

1	**Modulbezeichnung** 93330	**Deep Learning for Beginners** Deep learning for beginners	**2,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Deep Learning for Beginners (VHB-Kurs) (2 SWS)	2,5 ECTS
3	Lehrende	Aline Sindel	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Andreas Maier
5	**Inhalt**	Neural networks have had an enormous impact on research in image and signal processing in recent years. In this course, you will learn all the basics about deep learning in order to understand how neural network systems are built. The course is addressed to students who are new to the field. In the beginning of the course, we introduce you to the topic with some applications of deep learning in the field of medical imaging, digital humanities and industry projects. Before we dive into the core elements of neural networks, there are two lecture units on the fundamentals of signal and image processing to teach you relevant parts of system theory such as convolutions, Fourier transform, and sampling theorem. In the next lecture units, you learn the basic blocks of neural networks, such as backpropagation, fully connected layers, convolutional layers, activation functions, loss functions, optimization, and regularization strategies. Then, we look into common practices for training and evaluating neural networks. The next lecture unit is focusing on common neural network architectures, such as LeNet, Alexnet, and VGG. It follows a lecture unit about unsupervised learning that contains the principles of autoencoders and generative adversarial networks. Lastly, we cover some applications of deep learning in segmentation and object detection. The accompanying programming exercises will provide a deeper understanding of the workings and architecture of neural networks, in which you will develop a basic neural network from scratch in pure Python without using deep learning frameworks, such as PyTorch or TensorFlow. At the end of the semester, there will be a written exam.
6	**Lernziele und Kompetenzen**	The students • explain the different neural network components, • compare and analyze methods for optimization and regularization of neural networks, • compare and analyze different CNN architectures, • explain deep learning techniques for unsupervised / semi-supervised and weakly supervised learning, • explain different deep learning applications, • implement the presented methods in Python, • effectively investigate raw data, intermediate results and results of Deep Learning techniques on a computer, • autonomously supplement the mathematical foundations of the presented methods by self-guided study of the literature, • discuss the social impact of applications of deep learning applications.

7	**Voraussetzungen für die Teilnahme**	Requirements: mathematics for engineering, basic knowledge of python
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel (60 Minuten)
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 75 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 44150	**Diagnostic Medical Image Processing** Diagnostic medical image processing	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Medical Image Processing for Diagnostic Applications (VHB-Kurs) (4 SWS)	5 ECTS
3	Lehrende	Manuela Meier Luis Rivera Monroy Arpitha Ravi	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Andreas Maier
5	**Inhalt**	English version: The contents of the module comprise basics about medical imaging modalities and acquisition hardware. Furthermore, details on acquisition-dependent preprocessing are covered for image intensifiers, flat-panel detectors, and MR. The fundamentals of 3D reconstruction from parallel-beam to cone-beam reconstruction are also covered. In the last chapter, rigid registration for image fusion is explained. Deutsche Version: Die Inhalte des Moduls umfassen Grundlagen der medizinischen Bildverarbeitung und Aufnahmeprinzipien. Darüber hinaus werden Details der Vorverarbeitung für Bildverstärker, Flachpaneldetektoren und MR erklärt. Die Grundlagen der Rekonstruktion von Parallelstrahl bis hin zur Kegelstrahl-Tomographie werden ebenfalls behandelt. Im letzten Kapitel wird starre Registrierung für Bildfusion erläutert.
6	**Lernziele und Kompetenzen**	English Version: The participants • understand the challenges in interdisciplinary work between engineers and medical practitioners. • develop understanding of algorithms and math for diagnostic medical image processing. • learn that creative adaptation of known algorithms to new problems is key for their future career. • develop the ability to adapt algorithms to different problems. • are able to explain algorithms and concepts of the module to other engineers. Deutsche Version: Die Teilnehmenden • verstehen die Herausforderungen in der interdisziplinären Arbeit zwischen Ingenieuren und Ärzten. • entwickeln Verständnis für Algorithmen und Mathematik der diagnostischen medizinischen Bildverarbeitung. • erfahren, dass kreative Adaption von bekannten Algorithmen auf neue Probleme der Schlüssel für ihre berufliche Zukunft ist. • entwickeln die Fähigkeit Algorithmen auf verschiedene Probleme anzupassen. • sind in der Lage, Algorithmen und Konzepte des Moduls anderen Studierenden der Technischen Fakultät zu erklären.

7	**Voraussetzungen für die Teilnahme**	Ingenieurmathematik Engineering Mathematics
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	schriftlich/mündlich (60 Minuten)
11	**Berechnung der Modulnote**	schriftlich/mündlich (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 150 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 645618	**Human Computer Interaction** Human computer interaction	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Björn Eskofier Madeleine Flaucher
5	**Inhalt**	Das Modul vermittelt Konzepte, Prinzipien, Modelle, Methoden und Techniken für die effektive Entwicklung von benutzerfreundlichen Mensch-Computer-Schnittstellen. Das Thema moderner Benutzungsschnittstellen wird dabei für klassische Computer aber auch für mobile Geräte, eingebettete Systeme, Automobile und intelligente Umgebungen betrachtet. Die folgenden Themen werden im Modul behandelt: • Einführung in die Grundlagen der Mensch-Computer-Interaktion, historische Entwicklung • Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme • Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers • Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides • Ein- und Ausgabegeräte, Entwurfsraum für interaktive Systeme • Analyse-, Entwurfs- und Entwicklungsmethoden und -werkzeuge für Benutzungsschnittstellen • Prototypische Realisierung und Implementierung von interaktiven Systemen, Werkzeuge • Architekturen für interaktive Systeme, User Interface Toolkits und Komponenten • Akzeptanz, Evaluationsmethoden und Qualitätssicherung Contents: The module aims to teach basic knowledge of concepts, principles, models, methods and techniques for developing highly user-friendly Human-Computer Interfaces. Beyond traditional computer systems, modern user interfaces are also discussed in the context of automobile and intelligent environments, mobile devices and embedded systems. This module addresses the following topics: • Introduction to the basics of Human-Computer Interaction • Design principles and models for modern user interfaces and interactive systems • Information processing of humans, perception, motor skills, properties and skills of the users • Interaction concepts, metaphors, standards, norms and style guides • In- and output devices, design space for interactive systems

		• Analysis-, design- and development of methodologies and tools for easy-to-use user interfaces • Prototypic implementation of interactive systems • Architectures for interactive systems, User Interface Toolkits and components • Acceptance, evaluation methods and quality assurance
6	**Lernziele und Kompetenzen**	• Studierende entwickeln ein Verständnis für Modelle, Methoden und Konzepte der Mensch-Computer-Interaktion. • Sie lernen verschiedene Ansätze für den Entwurf, die Entwicklung und Bewertung von Benutzungsschnittstellen kennen und verstehen deren Vor- und Nachteile. • Die Teilnahme an der Veranstaltung versetzt Studierende in die Lage, einen Entwicklungsprozess in der Mensch-Computer-Interaktion zu verstehen und umzusetzen. • Sie werden weiterhin in die Lage versetzt, dies vor dem Hintergrund der Informationsverarbeitungsfähigkeit, Wahrnehmung und Motorik des Benutzers zu gestalten. • Passende Methoden der Evaluation sowie Akzeptanz- und Qualitätssicherung werden erlernt. Learning Objectives and Competences: • Students develop an understanding of models, methods and concepts in the field of Human-Computer Interaction. • They learn different approaches for designing, developing and evaluating User Interfaces and their advantages and disadvantages. • Joining the course enables students to understand and execute a development process in Human-Computer Interaction. • Students will be able to do a UI evaluation by learning the basics of information processing, perception and motoric skills of the user. • Appropriate evaluation methods, as well as acceptance and quality assurance aspects, will be learned.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Medieninformatik Bachelor of Science Informatik 20092 Vertiefungsrichtung Kommunikationssysteme Bachelor of Science Informatik 20222 Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur Electronic exam (in presence), 90min
11	**Berechnung der Modulnote**	Klausur (100%)

12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 44140	**Interventional Medical Image Processing** Interventional medical image processing	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Medical Image Processing for Interventional Applications (VHB-Kurs) (4 SWS)	5 ECTS
3	Lehrende	Manuela Meier Arpitha Ravi Luis Rivera Monroy	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Andreas Maier
5	**Inhalt**	English Version: This module focuses on recent developments in image processing driven by medical applications. All algorithms are motivated by practical problems. The mathematical tools required to solve the considered image processing tasks will be introduced. The module starts with an overview on preprocessing algorithms such as scatter correction for x-ray images, edge detection, super-resolution and edge-preserving noise reduction. The second chapter describes automatic image analysis using feature descriptors, key point detection, and segmentation using bottom-up algorithms such as the random walker or top-down approaches such as active shape models. Furthermore, the module covers geometric calibration algorithms for single view calibration, epipolar geometry, and factorization. The last part of the module covers non-rigid registration based on variational methods and motion-compensated image reconstruction. Deutsche Version: Das Modul ist auf die jüngsten Entwicklungen in der Verarbeitung von medizinischen Bildern ausgerichtet. Alle Algorithmen werden durch praktische Probleme motiviert. Die mathematischen Werkzeuge, die für die Bildverarbeitungsaufgaben benötigt werden, werden eingeführt. Das Modul beginnt mit einem Überblick über Vorverarbeitungsalgorithmen, wie zum Beispiel Streustrahlkorrektur für Röntgenbilder, Kantenerkennung, Superresolution und kantenerhaltende Rauschunterdrückung. Das zweite Kapitel beschreibt die automatische Bildanalyse mit Merkmalsdeskriptoren, Punkterkennung und Segmentierung mit Bottom-up-Algorithmen wie dem Random-Walker oder Top-Down-Ansätzen wie aktiven Formmodellen. Darüber hinaus deckt die Vorlesung auch geometrische Kalibrierungsalgorithmen zur Einzelansicht-Kalibrierung, Epipolargeometrie und Faktorisierung ab. Der letzte Teil des Moduls deckt nicht-starre Registrierung auf der Grundlage von Variationsmethoden und bewegungskompensierter Bildrekonstruktion ab.

6	**Lernziele und Kompetenzen**	English Version: The participants • summarize the contents of the lecture. • apply pre-processing algorithms such as scatter correction and edge-preserving filtering. • extract information from images automatically by image analysis methods such as key point detectors and segmentation algorithms. • calibrate projection geometries for single images and image sequences using the described methods. • develop non-rigid registration methods using variational calculus and different regularizers. • adopt algorithms to new domains by appropriate modifications. Deutsche Version: Die Teilnehmenden • fassen die Inhalte der Vorlesung zusammen. • wenden Vorverarbeitungsalgorithmen wie Streustrahlkorrektur und kantenerhaltende Filterung an. • extrahieren automatisch Informationen aus Bildern, indem sie Bildanalyseverfahren wie Punktdetektoren und Segmentierungsalgorithmen verwenden. • kalibrieren Projektionsgeometrien für Einzelbilder und Bildsequenzen mit den beschriebenen Methoden. • entwickeln nicht-starre Registrierungsmethoden mit Hilfe von Variationsrechnung und unterschiedlichen Regularisierern. • wenden Algorithmen auf neue Modalitäten durch entsprechende Änderungen im Algorithmus an.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	schriftlich/mündlich (60 Minuten)
11	**Berechnung der Modulnote**	schriftlich/mündlich (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 150 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 65718	**Introduction to Machine Learning** Introduction to machine learning	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Andreas Maier
5	**Inhalt**	Das Modul hat zum Ziel, die Studierenden mit dem prinzipiellen Aufbau eines Mustererkennungssystems vertraut zu machen. Es werden die einzelnen Schritte von der Aufnahme der Daten bis hin zur Klassifikation von Mustern erläutert. Das Modul beginnt dabei mit einer kurzen Einführung, bei der auch die verwendete Nomenklatur eingeführt wird. Die Analog-Digital-Wandlung wird vorgestellt, wobei der Schwerpunkt auf deren Auswirkungen auf die weitere Signalanalyse liegt.Im Anschluss werden gebräuchliche Methoden der Vorverarbeitung beschrieben. Ein wesentlicher Bestandteil eines Mustererkennungssystems ist die Merkmalsextraktion. Verschiedene Ansätze zur Merkmalsberechnung/-transformation werden gezeigt, darunter Momente, Hauptkomponentenanalyse und Lineare Diskriminanzanalyse.Darüber hinaus werden Möglichkeiten vorgestellt, Merkmalsrepäsentationen direkt aus den Daten zu lernen. Das Modul schließt mit einer Einführung in die maschinelle Klassifikation. In diesem Kontext wird der Bayes- und der Gauss-Klassifikator besprochen. The module aims to familiarize students with the basic structure of a pattern recognition system. The individual steps from the acquisition of data to the classification of patterns are explained. The module starts with a short introduction, which also introduces the used nomenclature. Analog-to-digital conversion is introduced, with emphasis on its impact on further signal analysis.Common methods of preprocessing are then described. An essential component of a pattern recognition system is feature extraction. Various approaches to feature computation/ transformation are demonstrated, including moments, principal component analysis, and linear discriminant analysis.In addition, ways to learn feature representations directly from the data are presented. The module concludes with an introduction to machine classification. In this context, the Bayes and Gauss classifiers are discussed. T
6	**Lernziele und Kompetenzen**	Die Studierenden • erklären die Stufen eines allgemeinen Mustererkennungssystems • verstehen Abtastung, das Abtasttheorem und Quantisierung • verstehen und implementieren Histogrammequalisierung und -dehnung • vergleichen verschiedene Schwellwertmethoden • verstehen lineare, verschiebungsinvariante Filter und Faltung

		• wenden verschiedene Tief- und Hochpassfilter sowie nichtlineare Filter an • wenden verschiedene Normierungsmethoden an • verstehen den Fluch der Dimensionalität • erklären verschiedene heuristische Merkmalsberechnungsmethoden, z.B. Projektion auf einen orthogonalen Basisraum, geometrische Momente, Merkmale basierend auf Filterung • verstehen analytische Merkmalsberechnungsmethoden, z.B. Hauptkomponentenanalyse, Lineare Diskriminanzanalyse • verstehen die Basis von Repräsentationslernen • erläutern die Grundlagen der statistischen Klassifikation (Bayes-Klassifikator) • benutzen die Programmiersprache Python, um die vorgestellten Verfahren der Mustererkennung anzuwenden • lernen praktische Anwendungen kennen und wenden die vorgestellten Algorithmen auf konkrete Probleme an The students • explain the stages of a general pattern recognition system • understand sampling, the sampling theorem, and quantization • understand and implement histogram equalization and expansion • compare different thresholding methods • understand linear, shift invariant filters and convolution • apply various low-pass, high-pass, and nonlinear filters • apply different normalization methods • understand the curse of dimensionality • explain different heuristic feature calculation methods, e.g. projection on an orthogonal base space, geometric moments, features based on filtering • understand analytical feature computation methods, e.g. principal component analysis, linear discriminant analysis • understand the basis of representation learning • explain the basics of statistical classification (Bayes classifier) • use the programming language Python to apply the presented pattern recognition methods • learn practical applications and apply the presented algorithms to concrete problems
7	**Voraussetzungen für die Teilnahme**	Ein Mustererkennungssystem besteht aus den folgenden Stufen: Aufnahme von Sensordaten, Vorverarbeitung, Merkmalsextraktion und maschinelle Klassifikation. Dieses Modul beschäftigt sich in erster Linie mit den ersten drei Stufen und schafft damit die Grundlage für weiterführende Module (Pattern Recognition und Pattern Analysis). A pattern recognition system consists of the following stages: Sensor Data Acquisition, Preprocessing, Feature Extraction, and Machine Classification. This module primarily deals with the first three stages and

		thus creates the basis for more advanced modules (Pattern Recognition and Pattern Analysis).
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Vorlesungsfolien/lecture slides • Heinrich Niemann: Klassifikation von Mustern, 2. überarbeitete Auflage, 2003 • Sergios Theodoridis, Konstantinos Koutroumbas: Pattern Recognition, 4. Auflage, Academic Press, Burlington, 2009 • Richard O. Duda, Peter E. Hart, David G. Stock: Pattern Classification, 2. Auflage, John Wiley & Sons, New York, 2001

1	**Modulbezeichnung** 93340	**Introduction to Network Science** Introduction to network science	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. David Blumenthal
5	**Inhalt**	Networks are fundamental data structures for modeling and analyzing complex biological, technological, or social systems. This course provides an introduction to the science of complex networks and their applications. The following topics will be covered: • Very brief introduction to graph theory, the mathematical underpinning of network science, • node centrality measures, • models of random networks, • network motifs, • degree correlations, • community detection, • network distance models, • evolving networks, • temporal networks.
6	**Lernziele und Kompetenzen**	Students will • get familiar with the basics of graph theory, • learn how to use networks to model complex relationships, • get familiar with the most important techniques for analyzing complex networks, • acquire hands-on experience in analyzing complex networks with the widely used Python library NetworkX.
7	**Voraussetzungen für die Teilnahme**	Since the lecture will be accompanied by programming exercises in Python, prior knowledge of this programming language is recommended. For students without prior experience, a very brief introduction to Python will be provided in the first two exercise sessions.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	A. Barabási and M. Pósfai, Network Science, Cambridge University Press, Cambridge, 2016, http://barabasi.com/networksciencebook/.

1	**Modulbezeichnung** 428256	**Maschinelles Lernen für Zeitreihen** Machine learning for time series	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Maschinelles Lernen für Zeitreihen (2 SWS)	2,5 ECTS
3	Lehrende	Dr. Dario Zanca Prof. Dr. Björn Eskofier	

4	**Modulverantwortliche/r**	Prof. Dr. Björn Eskofier Dr. Dario Zanca
5	**Inhalt**	Aim of the lecture is to teach Machine learning (ML) and Deep Learning (DL) methods for a variety of time series applications. The following topics will be covered: • Fundamentals and an overview of applications of time series analysis. • Fundamentals of ML methods, such as Gaussian processes, State Space models, and Autoregressive models for time series. • Design, implementation and evaluation of ML methods in order to address time series problems. • Advanced DL methods for time-series, such as Convolutional, Recurrent, and Attention-based models. • Working with widely-used toolboxes that can be used for implementation of ML methods, such as Tensorflow or PyTorch.
6	**Lernziele und Kompetenzen**	• Students can describe concepts of time series problems and their wide applications in industry, medicine, finance, etc. • Students can explain concepts of ML/DL methods in general and tackling time series problems in particular • Students understand the characteristics of time series data and are capable of developing and implementing ML/DL methods to model, predict and manipulate such data in concrete problems
7	**Voraussetzungen für die Teilnahme**	This is a specialisation lecture; successful completion of the lectures "IntroPR" and/or "Pattern Recognition" / "Pattern Analysis" is recommended. Concepts taught in "IntroPR" are assumed here as basic knowledge.
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222 electronic exam (remote), 90 min.
10	**Studien- und Prüfungsleistungen**	Variabel (90 Minuten)
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	nur im Wintersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Pattern recognition and machine learning. Christopher M. Bishop, Springer, 2006 • The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2009 • Machine Learning: A Probabilistic Perspective. Kevin Murphy, MIT press, 2012 • Deep Learning. Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016

1	**Modulbezeichnung** 639119	**Music Processing Analysis - Lecture and Exercise** Music processing analysis - Lecture and exercise	**5 ECTS**
2	Lehrveranstaltungen	Übung: Music Processing Analysis - Exercise (2 SWS) Vorlesung: Music Processing Analysis (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr. Meinard Müller	

4	**Modulverantwortliche/r**	Prof. Dr. Meinard Müller
5	**Inhalt**	Music signals possess specific acoustic and structural characteristics that are not shared by spoken language or audio signals from other domains. In fact, many music analysis tasks only become feasible by exploiting suitable music-specific assumptions. In this course, we study feature design principles that have been applied to music signals to account for the music-specific aspects. In particular, we discuss various musically expressive feature representations that refer to musical dimensions such as harmony, rhythm, timbre, or melody. Furthermore, we highlight the practical and musical relevance of these feature representations in the context of current music analysis and retrieval tasks. Here, our general goal is to show how the development of music-specific signal processing techniques is of fundamental importance for tackling otherwise infeasible music analysis problems.
6	**Lernziele und Kompetenzen**	Fachkompetenz Verstehen • Die Studierenden stellen zentrale Aufgabenstellungen der Musikverarbeitung in eigenen Worten dar und skizzieren Lösungsansätze. • Die Studierenden verstehen die Eigenschaften von unterschiedlichen Darstellungsformen von Musik. • Die Studierenden interpretieren Signaleigenschaften anhand von Visualisierungen (Exercise). Anwenden • Die Studierenden wenden grundlegende Algorithmen zur Analyse und zum Vergleich von Musiksignalen an. • Die Studierenden können voraussagen, wie sich unterschiedliche musikalische Eigenschaften bei der Signalanalyse auswirken. • Die Studierenden implementieren Algorithmen zur Analyse, zum Vergleich und zur inhaltsbasierten Suche von Musiksignalen (Exercise). Analysieren • Die Studierenden beobachten und diskutieren die Bedeutung und Auswirkung von Parametern bei der Musikanalyse. • Die Studierenden stellen unterschiedliche Verfahren bei der Analyse von Periodizitäten gegenüber. • Die Studierenden analysieren und erforschen Eigenschaften von Musiksignalen mittels automatisierter Methoden (Exercise). • Die Studierenden klassifieren und strukturieren Musikdaten mittels Lernverfahren (Exercise).

		Evaluieren (Beurteilen)
		• Die Studierenden hinterfragen Annahmen, die implizit bei der Verwendung von Analysemethoden gemacht werden.
		• Die Studierenden schätzen ein, wann Methoden bei der Analyse von gewissen Musiksignalen funktionieren könnten und wann sie typischerweise versagen.
		• Die Studierenden evaluieren automatisierte Methoden mittels geeigneter Evaluationsmaße unter Verwendung von manuell erstellten Annotationen (Exercise).
		Erschaffen
		•
		  ° Die Studierenden bereiten sich auf die Vorlesung anhand ausgewählter Literatur vor.
		  ° Die Studierenden hinterfragen bestehende Ansätze hinsichtlich ihrer Anwendbarkeit in der Praxis.
		  ° Die Studierenden beachten Fragen der Effizienz bei den diskutierten Algorithmen.
		  ° Die Studierenden entwickeln praktische Lösungswege für Problem in der Musikverarbeitung (Exercise)
		  ° Die Studierenden hinterfragen ihr Verständnis von dem Gelernten anhand von Übungsaufgaben.
		  ° Die Studierenden formulieren Fragen und stellen diese in der Vorlesung an den Dozenten und die Zuhörerschaft.
		  ° Die Studierenden nutzen Verbesserungshinweise des Betreuers und der Tutoren zur Verbesserung ihrer Lernstrategien (Exercise).
		  ° Die Studierenden organisieren selbständig Lerngruppen, in denen der Stoff diskutiert und vertieft wird.
		  ° Die Studierenden simulieren mit ihren Kommilitonen mündliche Prüfungen.
		  ° Die Studierenden entwickeln und implementieren Software im Team (Exercise).
		  ° Die Studierenden geben Kommilitonen im Rahmen ihrer Zusammenarbeit wertschätzendes Feedback (Exercise).

7	**Voraussetzungen für die Teilnahme**	In this course, we discuss a number of current research problems in music processing or music information retrieval (MIR) covering aspects from information science and digital signal processing. We provide the necessary background information and give numerous motivating examples so that no specialized knowledge is required. However, the students should have a solid mathematical background. The lecture is accompanied by readings from textbooks or the research literature. Furthermore, the students are required to experiment with the presented algorithms using Python.
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Medieninformatik Bachelor of Science Informatik 20092

		Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Meinard Müller Fundamentals of Music Processing Using Python and Jupyter Notebooks 2nd edition, 495 p., hardcover ISBN: 978-3-030-69807-2 Springer, 2021 http://www.music-processing.de/ https://www.audiolabs-erlangen.de/FMP

1	**Modulbezeichnung** 44130	**Pattern Recognition** Pattern recognition	**5 ECTS**
2	Lehrveranstaltungen	Übung: PR Exercise (1 SWS) Vorlesung: Pattern Recognition (3 SWS)	1,25 ECTS 3,75 ECTS
3	Lehrende	Siming Bayer Paul Stöwer Prof. Dr.-Ing. Andreas Maier	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Andreas Maier
5	**Inhalt**	Mathematical foundations of machine learning based on the following classification methods: • Bayesian classifier • Logistic Regression • Naive Bayes classifier • Discriminant Analysis • norms and norm dependent linear regression • Rosenblatt's Perceptron • unconstraint and constraint optimization • Support Vector Machines (SVM) • kernel methods • Expectation Maximization (EM) Algorithm and Gaussian Mixture Models (GMMs) • Independent Component Analysis (ICA) • Model Assessment • AdaBoost Mathematische Grundlagen der maschinellen Klassifikation am Beispiel folgender Klassifikatoren: • Bayes-Klassifikator • Logistische Regression • Naiver Bayes-Klassifikator • Diskriminanzanalyse • Normen und normabhängige Regression • Rosenblatts Perzeptron • Optimierung ohne und mit Nebenbedingungen • Support Vector Maschines (SVM) • Kernelmethoden • Expectation Maximization (EM)-Algorithmus und Gaußsche Mischverteilungen (GMMs) • Analyse durch unabhängige Komponenten • Modellbewertung • AdaBoost
6	**Lernziele und Kompetenzen**	Die Studierenden • verstehen die Struktur von Systemen zur maschinellen Klassifikation einfacher Muster • erläutern die mathematischen Grundlagen ausgewählter maschineller Klassifikatoren • wenden Klassifikatoren zur Lösung konkreter Klassifikationsproblem an

		• beurteilen unterschiedliche Klassifikatoren in Bezug auf ihre Eignung • verstehen in der Programmiersprache Python geschriebene Lösungen von Klassifikationsproblemen und Implementierungen von Klassifikatoren Students • understand the structure of machine learning systems for simple patterns • explain the mathematical foundations of selected machine learning techniques • apply classification techniques in order to solve given classification tasks • evaluate various classifiers with respect to their suitability to solve the given problem • understand solutions of classification problems and implementations of classifiers written in the programming language Python
7	**Voraussetzungen für die Teilnahme**	• Well grounded in probability calculus, linear algebra/matrix calculus • The attendance of our bachelor course 'Introduction to Pattern Recognition' is not required but certainly helpful. • Gute Kenntnisse in Wahrscheinlichkeitsrechnung und Linearer Algebra/Matrizenrechnung • Der Besuch der Bachelor-Vorlesung 'Introduction to Pattern Recognition' ist zwar keine Voraussetzung, aber sicherlich von Vorteil.
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch Englisch
16	**Literaturhinweise**	• Richard O. Duda, Peter E. Hart, David G. Stock: Pattern Classification, 2nd edition, John Wiley&Sons, New York, 2001 • Trevor Hastie, Robert Tobshirani, Jerome Friedman: The Elements of Statistical Learning - Data Mining, Inference, and Prediction, 2nd edition, Springer, New York, 2009

		• Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, New York, 2006

1	**Modulbezeichnung** 44455	**Speech and Language Processing**	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Andreas Maier Prof. Seung Hee Yang
5	**Inhalt**	Nach Behandlung der grundlegenden Mechanismen menschlicher Spracherzeugung und Sprachwahrnehmung gibt die Vorlesung eine detaillierte Einführung in (vornehmlich) statistisch orientierte Methoden der maschinellen Erkennung gesprochener Sprache. Schwerpunktthemen sind Merkmalgewinnung, Vektorquantisierung, akustische Sprachmodellierung mit Hilfe von Markovmodellen, linguistische Sprachmodellierung mit Hilfe stochastischer Grammatiken, prosodische Information sowie Suchalgorithmen zur Beschleunigung des Dekodiervorgangs. After focussing on of the basic mechanisms of human speech generation and speech perception the lecture gives a detailed introduction to (mainly) statistically oriented methods of automatic recognition of spoken language. Main topics are feature extraction, vector quantization, acoustic speech modeling with the help of Markov models, linguistic speech modeling with the help of stochastic grammars, prosodic information as well as search algorithms to speed up the decoding process.
6	**Lernziele und Kompetenzen**	Die Studierenden • verstehen die Grundlagen der menschlichen Sprachproduktion und die akustischen Eigenschaften unterschiedlicher Phonemklassen • erklären den allgemeinen Aufbau eines Mustererkennungssystems • verstehen Abtastung, das Abtasttheorem und Quantisierung in Bezug auf Sprachsignale • verstehen die Fourier-Transformation und mathematische Modelle der Sprachproduktion • verstehen harte und weiche Vektorquantisierungsmethoden • verstehen unüberwachtes Lernen (EM-Algorithmus) • verstehen Hidden Markov-Modelle (HMMs) • erklären stochastische Sprachmodelle The students • understand the principles of human speech production and acoustic properties of the different phoneme classes • explain the general pipeline of a pattern recognition system • understand sampling, the sampling theorem, and quantization w.r.t. speech signals

		• understand Fourier transformation and mathematical models of speech production • understand hard and soft vector quantization methods • understand unsupervised learning (EM-algorithm) • understand Hidden Markov Models (HMMs) • explain stochastic language models
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel (90 Minuten)
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Niemann H.: Klassifikation von Mustern; Springer, Berlin 1983 • Niemann H.: Pattern Analysis and Understanding; Springer, Berlin 1990 • Schukat-Talamazzini E.G.: Automatische Spracherkennung; Vieweg, Wiesbaden 1995 • º Rabiner L.R., Juang B.H.: Fundamentals of Speech Recognition; Prentice Hall, New Jersey 1993

Vertiefungsrichtung Programmiersysteme

1	**Modulbezeichnung** 44240	**Grundlagen des Übersetzerbaus** Foundations of compiler construction	**7,5 ECTS**
2	Lehrveranstaltungen	Übung: Übungen zu Grundlagen des Übersetzerbaus (2 SWS) Vorlesung: Grundlagen des Übersetzerbaus (2 SWS)	- 7,5 ECTS
3	Lehrende	Florian Mayer Sascha Hofmann Tobias Heineken Prof. Dr. Michael Philippsen	

4	**Modulverantwortliche/r**	Prof. Dr. Michael Philippsen
5	**Inhalt**	***[Deutsch:]*** Auf den ersten Blick erscheint es wenig sinnvoll, sich mit Übersetzerbau zu beschäftigen. Andere Themen scheinen wesentlich näher an der direkten Anwendbarkeit in der industriellen Praxis. Der erste Blick täuscht: • Übersetzer gehören wohl zu den am gründlichsten studierten mittelgroßen sequentiellen Software-Systemen. Man kann viel aus den Erfahrungen lernen, die im Laufe der Jahre gesammelt wurden. • In den Übungen, die die Vorlesung begleiten, werden Sie selbst einen (kleinen) Übersetzer entwickeln. • Für viele Teilnehmer wird dieses Projekt das erste größere Software-Projekt sein. Viele der Algorithmen aus dem Grundstudium werden angewendet. • Bei jedem von Ihnen verwendeten Übersetzer gehen Sie in der Regel davon aus, dass richtiger Coder erzeugt wird. In der Vorlesung erfahren Sie, wie das geforderte hohe Maß an Korrektheit und Zuverlässigkeit erreicht wird. • Sie erlangen ein Verständnis für Konzepte von Programmiersprachen und verstehen, welcher Maschinen-Code aus Sprachkonstrukten gemacht wird. Mit diesem Wissen im Hinterkopf verbessern Sie Ihre Fähigkeit, gute und effiziente Programme zu schreiben. • Übersetzer werden nicht nur für Programmiersprachen benötigt. Spezielle Übersetzer braucht man in vielen Bereichen des täglichen Informatik-Lebens z.B. zur Textformatierung, für Programmtransformationen, für aspektorientiertes Programmieren, für die Verarbeitung von XML, ... • Es gehört zu einer Ingenieur-Ausbildung, in der Lage zu sein, diejenigen Werkzeuge selbst zu fertigen, die man verwendet. Für Informatiker gehört daher ein Verständnis vom Innenleben eines Übersetzers zum Rüstzeug. **Fokus der Lehrveranstaltung:** Es werden Konzepte und Techniken der Übersetzerkonstruktion aus Sicht eines Übersetzerbauers und entlang der wesentlichen Arbeitsschritte eines Übersetzers (Frontend; Mittelschicht;

Backend) vorgestellt. Übungen und Praxisaufgaben ergänzen die Vorlesung. Hier entwickeln die Studierenden auf der Basis eines vorgegebenen Programmrahmens einen eigenen Übersetzer für die Programmiersprache e2, die speziell für den Übersetzerbau-Vorlesungszyklus entworfen wurde.

Behandelte Themenfelder:

- Prinzipien der Übersetzung imperativer Programmiersprachen
- Struktur eines Übersetzers
- Symbolentschlüssler (Scanner) und Zerteiler (Parser)
- Abstrakter Syntaxbaum (AST)
- Besuchermuster
- AST-Transformationen, Entzuckerung
- Symboltabellen und Sichtbarkeitsbereiche
- Semantische Analyse: Namensanalyse, Typprüfung
- Übersetzung von arithmetischen Ausdrücken und Kontrollflusskonstrukten in registerbasierte oder stapelbasierte Zwischensprachen
- Übersetzung von Methoden und Methodenaufrufen; Methodenschachteln
- Übersetzung objektorientierter Sprachen mit Einfachvererbung, Schnittstellen und Mehrfachvererbung
- Methodenauswahl in Java (überladene und überschriebene Methoden)
- Code-Generierung nach Sethi-Ullmann, Graham-Glanville, per Baumtransformation sowie mit Hilfe dynamischer Programmierung
- Registerallokation mit lokalen Techniken und mit Graphfärbung
- Instruktionsanordnung mit "list scheduling"
- Debugger

Themen der Vorlesungseinheiten:

1. Einführung (Überblick, modulare Struktur von Übersetzern, Frontend, Mittelschicht, Backend), Bootstrapping)
2. Symbolentschlüssler (Lexer) und Zerteiler (Parser), (Token, Literale, Symboltabelle, Grammatikklassen (LK(k), LL(k), ...), konkreter Syntaxbaum, Shift-Reduce-Parser)
3. AST und semantische Analyse (abstrakter Syntaxbaum, Besuchermuster, Double Dispatch, Sichtbarkeitsbereiche, Definitionstabelle)
4. Typkonsistenz (Typsicherheit, Typsystem, Typüberprüfung, Typberechnung, Typkonvertierung, attributierte Grammatiken)
5. AST-Transformationen (Transformationsschablonen für Ausdrücke, Transformation innerer und generischer Klassen)
6. Transformation in Zwischensprache (registerbasiert versus stapelbasiert, Übersetzung von arithmetischen Ausdrücken, Zuweisungen, mehrdimensionalen Feldern, struct-Datentypen und Kontrollflussstrukturen (einschließlich Kurzschlussauswertung))
7. Methodenschachteln und Kellerrahmen (relative Adressen, call by value/reference/name, geschachtelte Funktionen, Funktionszeiger,

Stack- und Framepointer, Funktionsaufruf, Prolog, Epilog)
8. Objektorientierte Sprachen I: Einfachvererbung (Symbol- und Typanalyse, Methodenauswahl mit Überschreiben und Überladen, virtuelle Methodenaufrufe, Klassendeskriptoren, dynamische Typprüfung und -wandlung)
9. Objektorientierte Sprachen II: Schnittstellen und Mehrfachvererbung (Interface v-Tables, dynamische Typprüfung und -wandlung mit Interfaces, Interfaces mit Default-Implementierung, Diamantenproblem)
10. Einfache Code-Erzeugung (Code-Selektion nach Sethi-Ullman, Register-Allokation, Instruktionsreihenfolge, optimale Code-Erzeugung für Ausdrucksbäume)
11. Fortgeschrittene Code-Erzeugung (Baumtransformation, Graham-Glanville, dynamisches Programmieren)
12. Registerallokation (Leistungsabschätzung, Lebendigkeitsintervalle, Kollisions- und Interferenzgraph, Spilling, Färbungsheuristiken, Aufteilung von Lebendigkeitsintervallen, 2nd Chance Bin Packing, Registerverschmelzung)
13. Parallelismus auf Instruktionsebene, Instruktionsreihenfolge, Debugger (Konflikte im Instruktionsfließband, List Scheduling, Delay-Slots, Sprungzielvorhersage, ptrace, Unterbrechungs- und Beobachtungspunkte, DWARF)

Meilensteine der Übungsbetriebs:
Im Rahmen der Übungen (siehe entsprechende Lehrveranstaltung) werden die in der Vorlesung vorgestellten Konzepte und Techniken zur Implementierung eines Übersetzers in die Praxis umgesetzt. Ziel der Übungen ist es, bis zum Ende des Semesters einen funktionsfähigen Übersetzer für die Beispiel-Programmiersprache e2 zu implementieren. Ein Rahmenprogramm ist gegeben, das in fünf Meilensteinen um selbstentwickelte Schlüsselkomponenten zu erweitern ist.
Folgende Meilensteine sind zu erreichen:
Meilenstein 1: Grammatik, AST-Konstruktion: Antlr-Produktionen, AST-Besucherschnittschelle, generischer AST-Besucher für return und Schleifen, AST-Besucher zur Visualisierung.
Meilenstein 2: Symbolanalyse, Symboltabelle, Standardfunktionen, AST-Besucher für die Symbolanalyse.
Meilenstein 3: Konstantenfaltung per AST-Transformation, Typanalyse mit bottom-up AST-Besuch, der implizite Typwandlungen bei Bedarf ergänzt.
Meilenstein 4: AST-Besucher zur Erzeugung der Zwischensprachrepräsentation, Übersetzung von arithmetischen Ausdrücken, return, Zuweisungen, logischen Ausdrücken, Bedingungen und Schleifen.
Meilenstein 5.0: Speicherzuteilung: Festlegung und Umsetzung der ABI Aufrufkonvention, Zuweisung von Speicheradressen zu Variablen; Kellerrahmenallokation; caller-save und callee-save Register.
Meilenstein 5.1: Code-Erzeugung: Implementierung der e2 Standardbibliothek; IR-Besucher zur Erzeugung von Assembly-Code.

Für die Meilensteine 1-3 soll der Übersetzer sowohl Integer- als auch Gleitkomma-Arithmetik unterstützen. Für die nachfolgenden Meilensteine reicht Integer-Arithmetik.

[English:]
The lecture teaches concepts and techniques of compiler construction from a compiler developer view, following the structure of the compiler frontend, middle end, and backend. Exercise sessions and practical assignments complement the lecture; the students implement their own compiler (based on a framework) for the e2 programming language, which is designed for this series of compiler construction lectures.
Content Summary

- Principles of compiling imperative programming languages
- Structure of a compiler
- Scanner and parser
- Abstract syntax trees (ASTs)
- Visitor design pattern
- AST transformations, desugaring
- Symbol tables and scopes
- Semantic analysis: name analysis, type checking
- Compilation of arithmetic expressions and control flow structures to register-based and stack-based intermediate languages
- Compilation of functions and function calls, activation records
- Compilation of object-oriented languages with single inheritance, interfaces, and multiple inheritance
- Method resolution in Java (overloaded and overridden methods)
- Code generation with Sethi-Ullmann algorithm, Graham-Glanville algorithm, tree transformations, and dynamic programming
- Register allocation with local techniques and graph coloring
- Instruction scheduling with the list scheduling technique
- Debuggers

Lecture Topics

- 1. Introduction: Class overview, modular structure of compilers (front-, middle-, and backend), compilation bootstrapping
- 2. Lexer and Parser: Tokens, literals, symbol table, grammar classes (LR(k), LL(k), ...), concrete syntax tree, shift-reduce parser
- 3. ASTs and semantic analysis: Abstract syntax tree, visitor pattern, double dispatch, scopes, definition table
- 4. Type consistency: Type safety, type system, type checks, type inference, type conversions, attributed grammars
- 5. AST transformations: Transformation patterns (arithmetics), transformation of nested and generic classes
- 6. Intermediate representations: Types of IRs, arithmetic operations, assignments, multidimensional array access, structs, control flow instructions, short-circuit evaluation

- 7. Activation record and stack frame: Relative addresses, call by value/reference/name, nested functions, function pointers, stack pointer and frame pointer, function calls: prolog and epilog
- 8. Object-oriented languages: single inheritance: Symbol and type analysis, method selection with method overloading and overriding, virtual method calls, class descriptors, dynamic type checks and casts
- 9. Object-oriented languages II: interfaces, multiple inheritance: Interface v-tables, dynamic type checks and casts with interfaces, interfaces with default implementations and state, diamond problem, virtual inheritance
- 10. Basic code generation: Code selection, register allocation, instruction order, basic blocks, optimal code generation for expression trees
- 11. Optimized code selection: Code selection as tree transformation, Graham-Glanville code generators, dynamic programming
- 12. Optimized register allocation: Performance approximations, liveness analysis, collision and interference graph, register spilling, coloring heuristics, optimistic extension, live range splitting, register coalescing, data structures
- 13. Instruction level parallelism, instruction order, debugger: Data, structural, and control conflicts in CPU pipelines, list scheduling, delay slots, branch predictions, superscalar and VLIW architectures, ptrace, break- and watch-points, DWARF

Assignment Milestones

For the assignments of this course, the students put the concepts and techniques presented in the lecture for implementing a compiler into practice. The goal of the assignments is to implement a functional compiler for the e2 programming language by the end of the semester. The e2 language is specifically designed for educational purposes; the students obtain a description of the language.

A framework for the implementation is provided to the students. The students implement the core components of the compiler in five milestones.

All milestones need to be fulfilled to pass the module; the last milestone contains two tasks. In particular, the milestones are:

- Milestone 1: Grammar definition and construction of the AST: ANTLR productions, AST visitor interface, and generic AST visitor for array accesses and return and loop statements; AST visitor for AST visualization.
- Milestone 2: Name analysis: symbol table; declaring standard functions; AST visitor for name analysis.
- Milestone 3: Constant folding and type analysis: AST transformations for constant folding; AST visitor for bottom-up type analysis, adding AST nodes for implicit casts;

		• Milestone 4: AST translation to intermediate representation: AST visitor to generate IR; translation of arithmetic, return, and assign statements, logical expressions, conditions, loops. • Milestone 5.0: Memory assignment: definition and implementation of the ABI calling convention; memory assignment of variables; stack frame allocation; caller-save and callee-save registers. • Milestone 5.1: Code generation: implementation of the e2 standard library; IR visitor to generate assembly code. For milestones one through three, the compiler needs to support both integer and floating-point arithmetic. For the last two milestones, only integer arithmetic is required.
6	**Lernziele und Kompetenzen**	***[Deutsch:]*** Die Studierenden • nennen die typischen Aufgaben und Datenstrukturen eines Übersetzers • erläutern das Konzept des Bootstrapping • beschreiben Struktur und Arbeitsweise eines Abtasters (Scanner) und zeigen Grenzen und Problemfälle auf • wenden Grammatiken zur Konstruktion von Zerteilern (Parser) an • kennen die Komplexität eines Zerteilers für Java • beschreiben die wichtigsten Aufgaben der semantischen Analyse und wenden diese am Beispiel verschiedener Programmiersprachen (insbesondere Java) an • skizzieren typische AST-Transformationen am Beispiel des Java-Übersetzers • veranschaulichen die Grundzüge der Java-Kellermaschine und die zugehörige Transformation von Quell- zu Byte-Code • analysieren die Unterschiede zwischen Programmiersprachen hinsichtlich Felder und Verbund-Strukturen • erläutern die Verwendung von Stapel- und Kellerspeicher bei der Programmausführung • kennen verschiedene Maschineninstruktionssätze • optimieren die Registerverwendung vor der Generierung von Maschinencode • wenden das Verfahren von Graham & Glanville zur Erzeugung von Maschinencode an • erkennen Grenzen der Optimierung bei der Code-Generierung und analysieren alternative Strategien • beschreiben den Unterschied zwischen statischer und dynamischer Ablaufplanung • untersuchen Besonderheiten des Übersetzerbaus für objekt-orientierte Sprachen • ergänzen einen vorgegebenen Abtaster und abstrakten Syntaxbaum, um alle Sprachkonstrukte einer Beispielsprache zu unterstützen

- implementieren Konstantenfaltung, den Aufbau der Symboltabelle und Typprüfung auf dem abstrakten Syntaxbaum
- erzeugen Zwischencode aus dem abstrakten Syntaxbaum
- bilden Kontrollstrukturen auf Sprünge ab
- veranschaulichen die Adressierung von (mehrdimensionalen) Feldern
- entwickeln Konventionen für Funktionsaufrufe und den Aufbau des Stacks
- berechnen Offsets fuer Variablen auf dem Stack.
- implementieren eine einfache Registervergabe.
- kennen Details verschiedener Prozessorarchitekturen
- generieren Maschinencode für mindestens eine Prozessorarchitektur
- implementieren eine Laufzeitbibliothek
- wenden Debugging für maschinennahen Code an

[English:]

Students who have successfully completed the module will have the ability to

- identify the components and data structures of a compiler
- explain the concept of bootstrapping
- describe the structure and operation of a lexer and show limitations and problem cases
- use grammars for the construction of parsers
- know the complexity of Java parsers
- describe the main tasks of semantic analysis and apply them to different programming languages (especially Java)
- outline typical AST transformations using the Java compiler as an example
- illustrate the basic features of the Java Virtual Machine (JVM) and the corresponding transformation from source to byte code
- analyze the differences between programming languages in terms of arrays and compound structures
- explain the use of stack memory in program execution
- know different machine instruction sets
- optimize register allocation before generating machine code
- apply the Graham-Glanville algorithm to generate machine code
- recognize limitations of optimization in code generation and to analyze alternative strategies
- describe the difference between static and dynamic scheduling
- examine features of compiler construction for object-oriented languages
- augment a given lexer and abstract syntax tree to support all language constructs in an example language
- implement constant folding, symbol table construction, and type checking on the abstract syntax tree
- generate intermediate code from the abstract syntax tree

		• map control structures to jumps • translate compound boolean expressions with shortcut evaluation • illustrate addressing of (multidimensional) arrays • design conventions for function calls and stack frame layout • calculate offsets for stack variables • implement a basic register allocation. • know details of different processor architectures • generate machine code for at least one processor architecture • implement a runtime library • apply debugging to machine code
7	**Voraussetzungen für die Teilnahme**	Participants of this lecture are expected to have profound skills in the following programming languages: • Java (assignments are implemented in Java) • Assembler
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Programmiersysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 50 h Eigenstudium: 175 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Modern Compiler Implementation in Java", A.W. Appel, Cambridge University Press, 1998 • Compilers - Principles, Techniques and Tools", A. Aho, R. Sethi, J. Ullmann, Addison-Wesley, 2006 • Modern Compiler Design", D. Grune, H. Bal, C. Jacobs, Langendoen, Wiley, 2002

1	**Modulbezeichnung** 505241	**Mainframe Programmierung** Mainframe programming	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Mainframe Programmierung I (0 SWS)	5 ECTS
3	Lehrende	Sebastian Wind	

4	**Modulverantwortliche/r**	Sebastian Wind
5	**Inhalt**	Der Begriff "Mainframe" bezeichnet grosse Rechenanlage, wie sie in der Wirtschaft für extrem grossen Anwendungen eingesetzt werden. Typische Branchen sind Banken und Versicherungen, aber auch Automobilhersteller und AI-Anwender. Der Online-Kurs soll nun die Möglichkeit eröffnen, Erfahrungen mit der Programmierung eines Mainframes zu sammeln. Dazu gehören die elementaren Programmieraufgaben wie editieren, übersetzen, binden, laden, ausführen und debuggen, die anhand von Beispielen in der Programmiersprache CoBOL geübt werden. Die Architektur der Mainframes werden sowohl aus Sicht der Rechnerarchitektur wie auch der Anwendersicht beleuchtet. Insbesondere werden die Virtualisierungsmöglichkeiten udn die gängigen Betriebssysteme wie z/OS und Linux auf den Mainframes behandelt. Den Abschluss und Ausblick bildet die Datenhaltung und die Integration in die IT-Systemlandschaft. Inhalt: 0. Begrüßung und Einführung 1. CoBOL Programmierung 2. Einführung Mainframes 3. IBM Mainframe Architektur 4. z/OS 5. Anwendungsprogrammierung 6. Virtualisierung 7. Linux 8. Integration in die IT-Systemlandschaft
6	**Lernziele und Kompetenzen**	Folgende Kompetenzen werden von den Studierenden erlernt: Fachkompetenz Wissen Kenntnis der wesentlichen Begriffe der Mainframe Programmierung Kenntniss der Programmierparadigmen Identizieren der Vor- und Nachteile des Einsatzes von Mainframes Verstehen Erläutern der Unterschiede zwischen konventioneller und kommerzieller DatenverabeitungSliziieren eines Mainframe-Einsatz-Szenarios Anwenden Anwendung der Programmierwerkzeuge für Mainframes Beherschung der wichtiges Kommandos des Mainframe-OS Organisation der Daten Analysieren Analyse und Konzeption einer Mainframe-Anwendung Evaluieren (Beurteilen)

		Evaluation einer bestehenden Architektur bezüglich der wirtschaftlichen, technologoischen und sozialen Aspekte
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222 Vertiefungsrichtung Programmiersysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich (90 Minuten)
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 150 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Wird über StudOn zur Verfügung gestellt.

1	**Modulbezeichnung** 93182	**Mainframe Programmierung II** Mainframe programming II	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Sebastian Wind
5	**Inhalt**	Aufbauend auf den Grundlagen der CoBOL-Programmierung und der Bedienung des zOS-Betriebssystems (aus dem Kurs Mainframe Programmierung I) werden nun komplexere Programmierkonstrukte und der Einsatz weiterer Werkzeuge behandelt. Zunächst werden neue Konstrukte zur Datenbehandlung in CoBOL vorgestellt, inkl. Zeichenketten und Dateibehandlung. JCL rundet dann den Progranmnierungsabschnitt ab. Die Datenhaltung geschieht entweder in Dateien oder in Datenbanken, weshalb sich ein Abschnitt über DB, SQL und die Verbindung zu JCL anschliesst. Ein Ausflug zu zLinux zeigt den Einsatz alternativer Betriebssysteme auf der z-Plattform. Die verschiedenen Varianten der Transaktionsverarbeitung stellt das darauf folgende Kapitel dar, u.a. CICS, Liberty, WebSphere, Message Queues, MQ und z/OS-Connect. Den Abschluss bildet ein Ausblick auf aktuelle Trends, z.B. Blockchains.
6	**Lernziele und Kompetenzen**	Folgende Kompetenzen werden von den Studierenden erlernt: Fachkompetenz Analysieren Lernende können ein Problem aus dem bereich Enterprose Computing in einzelne Teile zerlegen und so die Struktur des Problems verstehen. Sie können Widersprüche aufdecken, Zusammenhänge erkennen und Folgerungen ableiten und zwischen Fakten und Interpretationen unterscheiden. Lern- bzw. Methodenkompetenz Das Modul vermittelt sowohl Kompetenzen im selbstorganisierten Lernen, wie auch Erfahrungen mit einer multi-modalen Lernumgebung.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222 Vertiefungsrichtung Programmiersysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich (30 Minuten)
11	**Berechnung der Modulnote**	mündlich (100%)

12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 150 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Wird über StudOn verteilt.

Vertiefungsrichtung Rechnerarchitektur

1	**Modulbezeichnung** 44460	**Architekturen von Superrechnern** Architectures of supercomputers	**5 ECTS**
2	Lehrveranstaltungen	Übung: Exercises Architectures of Supercomputers / Übungen Architekturen von Superrechnern (2 SWS) Vorlesung: ArchSup (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr.-Ing. Dietmar Fey Farhad Ebrahimiazandaryani	

4	**Modulverantwortliche/r**	
5	**Inhalt**	• Principles of computer and processor architectures • Modern processor architectures • Homogeneous and heterogeneous multi/many-core processors • Parallel computer architectures • Classification and principles of coupling parallel computers • High speed networks in supercomputers • Examples of supercomputers • Programming of supercomputers
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Lernende können die Funktionsweise moderner in Superrechnern eingesetzter Prozessoren wiedergeben. Sie erkennen die besonderen Probleme im Zusammenahng mit dem Ernergieverbrauch und der Programmierung von Superrechnern. Verstehen Lernende können die Unterschiede bei der Kopplung paralleler Prozesse darstellen. Sie können Parallelrechner hinsichtlich ihrer Speicheranbindung und den grundlegenden Verarbeitungsprinzipien klassifizieren. Anwenden Lernenden sind in der Lage ein eigenes technisches oder mathematisches Problem zur Lösung auf einem Supercomputer umzusetzen und auszuführen. Anhand der in der Vorlesung gezeigten Beispiele sind sie in der Lage, Herausforderungen beim Auffinden von Flaschenhälsen zu verallgemeinern und für ihr konkretes Problem anzuwenden. Analysieren Lernende sind in der Lage, ihre Problemstellungen, z.B. naturwissenschaftliche oder technische Simulationsexperimente, hinsichtlich der Rechen- und Speicheranforderungen für einen Supercomputer geeignet für die Architekur zu charakterisieren. Evaluieren (Beurteilen) Lernende können mithilfe der aufgezeigten Methodiken zur Leistungsmesung von Parallerechnern unterschiedliche Rechnerarchitekturen bewerten und für ihre Problemstellung die passende Architektur auswählen.

7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Rechnerarchitektur Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	schriftlich (30 Minuten)
11	**Berechnung der Modulnote**	schriftlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 120 h Eigenstudium: 30 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 211243	**CPU Entwurf mit VHDL (CPU)** CPU Design with VHDL (CPU)	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Dietmar Fey
5	**Inhalt**	Die Vorlesung wird einen Einblick in die wichtigsten Strukturen eines Rechners und insbesondere die der CPU geben. Wie sind die Register, die ALU, die Caches usw. aufgebaut, wie sind sie miteinander verschaltet? Wie werden Pipelines gebildet? Um die Vorlesung verstehen und die Übungsaufgaben lösen zu können, werden VHDL-Kenntnisse benötigt. Diese werden in der Vorlesung vermittelt. Im ersten Teil der Vorlesung geht es darum, aus einfachen Und-, Oder- und Inverter-Gattern größere Strukturen aufzubauen. So werden bespielsweise die Register und die Komponenten der ALU (Addierer, Multiplizierer, Shifter usw.) nach und nach aufgebaut. Aus den Einzelteilen lassen sich dann wiederum die Registerbänke, die ALU, die Speichereinheit usw. zusammensetzen. Mit guten Rechenwerken allein ist eine CPU noch nicht wirklich schnell. Es müssen schnell genug Instruktionen gelesen und dekodiert sowie die Operanden geholt werden können. Nach dem schnellen Verarbeiten in den Rechenwerken müssen die Ergebnisse dann wiederum schnell in die Zielregister bzw. Ziel-Speicherzellen kopiert werden. Das wichtigste Verfahren, um hierbei Performance zu gewinnen, ist das Pipelining. Leider gibt es eine Reihe von Bedingungen, die erfüllt sein müssen, dass Pipelining wirklich gut funktioniert. So dürfen keine Daten-Abhängigkeiten im Code vorhanden sein, müssen Sprungziele rechtzeitig bekannt sein und dürfen bestimmte Ressourcen nicht mehrfach gleichzeitig genutzt werden. Mittels Forwarding, Sprungvorhersage und Caches können die Probleme gemildert werden. Mit den so gewonnenen Fähigkeiten sind dann auch Multi-Core- und Multi-Threading-Architekturen verständlich.
6	**Lernziele und Kompetenzen**	Die Studierenden verstehen den inneren Aufbau moderner CPUs (Pipeline, Registerbänke, ALU, Caches, Memory-Management-Unit, Segmentierungseinheit, ...) und können selbst performante CPUs aus einfachen Basis-Schaltungen aufbauen.
7	**Voraussetzungen für die Teilnahme**	keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Rechnerarchitektur Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich

11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 436348	**CPU Entwurf mit VHDL (Schwerpunkt VHDL)** CPU Design with VHDL (Focus on VHDL)	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Dietmar Fey
5	**Inhalt**	Die Vorlesung wird einen Einblick in die wichtigsten Strukturen eines Rechners und insbesondere die der CPU geben. Wie sind die Register, die ALU, die Caches usw. aufgebaut, wie sind sie miteinander verschaltet? Wie werden Pipelines gebildet? Um die Vorlesung verstehen und die Übungsaufgaben lösen zu können, werden VHDL-Kenntnisse benötigt. Diese werden in der Vorlesung vermittelt. Im ersten Teil der Vorlesung geht es darum, aus einfachen Und-, Oder- und Inverter-Gattern größere Strukturen aufzubauen. So werden bespielsweise die Register und die Komponenten der ALU (Addierer, Multiplizierer, Shifter usw.) nach und nach aufgebaut. Aus den Einzelteilen lassen sich dann wiederum die Registerbänke, die ALU, die Speichereinheit usw. zusammensetzen. Mit guten Rechenwerken allein ist eine CPU noch nicht wirklich schnell. Es müssen schnell genug Instruktionen gelesen und dekodiert sowie die Operanden geholt werden können. Nach dem schnellen Verarbeiten in den Rechenwerken müssen die Ergebnisse dann wiederum schnell in die Zielregister bzw. Ziel-Speicherzellen kopiert werden. Das wichtigste Verfahren, um hierbei Performance zu gewinnen, ist das Pipelining. Leider gibt es eine Reihe von Bedingungen, die erfüllt sein müssen, dass Pipelining wirklich gut funktioniert. So dürfen keine Daten-Abhängigkeiten im Code vorhanden sein, müssen Sprungziele rechtzeitig bekannt sein und dürfen bestimmte Ressourcen nicht mehrfach gleichzeitig genutzt werden. Mittels Forwarding, Sprungvorhersage und Caches können die Probleme gemildert werden. Mit den so gewonnenen Fähigkeiten sind dann auch Multi-Core- und Multi-Threading-Architekturen verständlich.
6	**Lernziele und Kompetenzen**	Die Studierenden verstehen den inneren Aufbau moderner CPUs (Pipeline, Registerbänke, ALU, Caches, Memory-Management-Unit, Segmentierungseinheit, ...) und können selbst performante CPUs aus einfachen Basis-Schaltungen aufbauen.
7	**Voraussetzungen für die Teilnahme**	keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Rechnerarchitektur Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich

11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 275245	**Heterogene Rechnerarchitekturen Online** Heterogeneous computing architectures online	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Heterogene Rechnerarchitekturen Online (0 SWS)	-
3	Lehrende	Prof. Dr.-Ing. Dietmar Fey Johannes Kliemt	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Dietmar Fey
5	**Inhalt**	Whereas heterogeneous architectures and parallel computing has filled an academic niche in the past it has become now a commodity technique with the rising of multi-core processors and programmable graphic cards. Even FPGAs play a role hereby in a certain extent due to their increasing importance as accelerator hardware what is clearly observable in the scientific community. However, on one side parallel hardware like multi-core and GPUs are now available nearly for everybody and not only for a selected selection of people, who have access to a parallel supercomputer. On the other side the knowledge about programming of this commodity hardware, and we mean here in particular hardware-orientated programming in order to squeeze out all offered GFlops and TFlops of such hardware, is still missing as well as the knowledge about the architecture details. To overcome this lack we offer this course HETRON. The e-learning course HETRON for the exploitation of parallel and heterogeneous computer architectures) focuses on two main topics which are closely related to each other. This concerns on one side the benefits of using different kinds of multi-core processors and parallel architectures built-up on base of these multicore processors. These architectures differ among each other in the number and in the complexity of its single processing nodes. We distinguish between systems consisting of a large number of simpler, so called fine-grained, processor cores vs. systems consisting of a smaller number of more complex, so called coarse-grained, processor cores. On the other side we lay our focus on that we want to do with these different heterogeneous parallel architectures, namely the execution of parallel programs. Of course this requires the use of parallel programming languages and environments, like CUDA or OpenMP. However, besides these questions of using the right syntax and the right compiler switches to optimize a parallel program it is a pre-requisite to understand how

		parallel computing really works. This refers (i) to the comprehension which basic mechanisms of parallel computing exist, (ii) where are the limits of getting more performance with parallel computing and (iii) in what context stand these mechanisms to heterogeneous architectures. In other words it handles the question which architecture is the best one for a certain parallelization technique. To teach these three topics, is one main goal we pursuit with the course HETRON, and of course, this more fundamental basics of heterogeneous and parallel computing have to be proven by means of concrete application examples to deepen the acquired knowledge about heterogeneous architectures and parallel computing principles.
6	**Lernziele und Kompetenzen**	Die Studierenden ... ...verstehen die Notwendigkeit sowie grundlegende Anwendungsfälle für heterogene Rechnerarchitekturen. ...können den grundlegenden Aufbau und das Zusammenspiel der Komponenten heterogener Rechnerarchitekturen erklären. ...erläutern grundsätzliche Parallelisierungsprinzipien wie Amdahls Law, High-Performance- und High-Througput-Computing sowie Parallelisierungsstrategien. ...können einfache Programme mit Hilfe der vermittelten Prallelisierungsprinzipien (Amdahls Law, High-Performance- und High-Throughput-Computing) analysieren und entsprechende Parallelisierungsstrategien entwickeln. ...erklären den Aufbau sowie Stärken und Schwächen von verschiedenen Architekturen wie CPUs, GPUs, Many-Core Prozessoren und FPGAs. ...implementieren ausgewählte Anwendungsbeispiele (SHA256 Algorithmus, Ising-Modell und Fast-Fourier-Transformation) auf oben genannte Architekturen. ...erforschen und bewerten verschiedener Parallelsierungstechniken in Abhängigkeit der Anwendung und der Architektur. ...erläutern die Grundlagen des Grid- und Cloud-Computings ...sind in der Lage parallele Berechnungen (SHA256) im Grid umzusetzen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Rechnerarchitektur Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)

12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 150 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 798810	**Rechnerarchitektur** Computer architecture	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Rechnerarchitektur (2 SWS) Übung: Ü RA (2 SWS) Übung: RÜ RA (2 SWS)	2,5 ECTS 2,5 ECTS 2,5 ECTS
3	Lehrende	Tobias Baumeister Prof. Dr.-Ing. Dietmar Fey Kenan Gündogan Philipp Gündisch	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Dietmar Fey
5	**Inhalt**	Die Vorlesung baut auf die in den Grundlagen der Rechnerarchitektur und -organisation vermittelten Inhalte auf und setzt diese mit weiterführenden Themen fort. Es werden zunächst grundlegende fortgeschrittene Techniken bei Pipelineverarbeitung und Cachezugriffen in modernen Prozessoren und Parallelrechnern behandelt. Ferner wird die Architektur von Spezialprozessoren, z.B. DSPs und Embedded Prozessoren behandelt. Es wird aufgezeigt, wie diese Techniken in konkreten Architekturen (Intel Nehalem, GPGPU, Cell BE, TMS320 DSP, Embedded Prozessor ZPU) verwendet werden. Zur Vorlesung wird eine Tafelübung angeboten. Mit erfolgreicher mündlicher Prüfung können 5 ECTS erworben werden. In den Tafelübungen werden die in der Vorlesung vermittelten Techniken durch zu lösende Aufgaben vertieft. In der Rechnerübung soll u.a. ein einfacher Vielkern-Prozessor auf Basis des ZPU-Prozessors mit Simulationswerkzeugen aufgebaut werden. Im Einzelnen werden folgende Themen behandelt: • Organisationsaspekte von CISC und RISC-Prozessoren • Behandlung von Hazards in Pipelines • Fortgeschrittene Techniken der dynamischen Sprungvorhersage • Fortgeschritten Cachetechniken, Cache-Kohärenz • Ausnutzen von Cacheeffekten • Architekturen von Digitalen Signalprozessoren • Architekturen homogener und heterogener Multikern-Prozessoren (Intel Corei7, Nvidia GPUs, RISC-V) • Architektur von Parallelrechnern (Clusterrechner, Superrechner) • Effiziente Hardware-nahe Programmierung von Mulitkern-Prozessoren (OpenMP, SSE, CUDA) • Leistungsmodellierung und -analyse von Multikern-Prozessoren (Roofline-Modell)
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Lernende können Wissen abrufen und wiedergeben. Sie kennen konkrete Einzelheiten wie Begriffe, Definitionen, Fakten, und Abläufe in einem Prozessor darlegen. Verstehen

		Lernende können Beispiele für Rechnerarchitekturen anführen, sie sind in der Lage, Schaubilder von Prozessoren zu interpretieren und die Abläufe in eigenen Worten zu beschreiben. Anwenden Lernende können beim Erstellen eigener Programme durch Transfer des Wissens über Interna von Prozesorarchitekturen Optimierungen hinsichtlich des Laufzeitverhaltens vornehmen. Analysieren Lernende können zwischen verschiedenen Varianten von Lösungen einer Prozessorarchitektur klassifizieren, die Gründe für durchgeführte Entwurfsentscheidungen erschließen, Unterscheide gegenüberstellen und gegeneinander bewerten. Lern- bzw. Methodenkompetenz Lernende erwerben die Fähigkeit selbstständig Testprogramme zum Bewerten der Leistungsfähigkeit eines Prozessors zu erstellen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Rechnerarchitektur Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio (90 Minuten)
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	• Patterson/Hennessy: Computer Organization und Design • Hennessy/Patterson: Computer Architecture - A Quantitative Approach • Stallings: Computer Organization and Architecture • Märtin: Rechnerarchitekturen

1	**Modulbezeichnung** 333815	**Rechnerarchitektur (Vorlesung mit Übung und Rechnerübung)** Computer architecture	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Rechnerarchitektur (2 SWS) Übung: Ü RA (2 SWS) Übung: RÜ RA (2 SWS)	2,5 ECTS 2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr.-Ing. Dietmar Fey Tobias Baumeister Philipp Gündisch Kenan Gündogan	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Dietmar Fey
5	**Inhalt**	Die Vorlesung baut auf die in den Grundlagen der Rechnerarchitektur und -organisation vermittelten Inhalte auf und setzt diese mit weiterführenden Themen fort. Es werden zunächst grundlegende fortgeschrittene Techniken bei Pipelineverarbeitung und Cachezugriffen in modernen Prozessoren und Parallelrechnern behandelt. Ferner wird die Architektur von Spezialprozessoren, z.B. DSPs und Embedded Prozessoren behandelt. Es wird aufgezeigt, wie diese Techniken in konkreten Architekturen (Intel Nehalem, GPGPU, Cell BE, TMS320 DSP, Embedded Prozessor ZPU) verwendet werden. Zur Vorlesung werden eine Tafel- und eine Rechnerübung angeboten. Die Rechnerübung erfodert 11 erfolgreich abgeschlossene Übungsaufgaben, diese gehen mit 10% in die Modulgesamtnote ein. Die verbleibenden 90% werden durch die mündliche Prüfung bestimmt. Insgesamt werden 7,5 ECTS erworben. In den Tafelübungen werden die in der Vorlesung vermittelten Techniken durch zu lösende Aufgaben vertieft. In der Rechnerübung soll u.a. ein einfacher Vielkern-Prozessor auf Basis des ZPU-Prozessors mit Simulationswerkzeugen aufgebaut werden. Im Einzelnen werden folgende Themen behandelt: • Organisationsaspekte von CISC und RISC-Prozessoren • Behandlung von Hazards in Pipelines • Fortgeschrittene Techniken der dynamischen Sprungvorhersage • Fortgeschritten Cachetechniken, Cache-Kohärenz • Ausnutzen von Cacheeffekten • Architekturen von Digitalen Signalprozessoren • Architekturen homogener und heterogener Multikern-Prozessoren (Intel Corei7, Nvidia GPUs, RISC-V) • Architektur von Parallelrechnern (Clusterrechner, Superrechner) • Effiziente Hardware-nahe Programmierung von Mulitkern-Prozessoren (OpenMP, SSE, CUDA) • Leistungsmodellierung und -analyse von Multikern-Prozessoren (Roofline-Modell)
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen

		Lernende können Wissen abrufen und wiedergeben. Sie können konkrete Einzelheiten wie Begriffe, Definitionen, Fakten, und Abläufe in einem Prozessor darlegen. Verstehen Lernende können Beispiele für Rechnerarchitekturen anführen, sie sind in der Lage, Schaubilder von Prozessoren zu interpretieren und die Abläufe in eigenen Worten zu beschreiben. Anwenden Lernende können beim Erstellen eigener Programme durch Transfer des Wissens über Interna von Prozesorarchitekturen Optimierungen hinsichtlich des Laufzeitverhaltens vornehmen. Analysieren Lernende können zwischen verschiedenen Varianten von Lösungen einer Prozessorarchitektur klassifizieren, die Gründe für durchgeführte Entwurfsentscheidungen erschließen, Unterscheide gegenüberstellen und gegeneinander bewerten. Lern- bzw. Methodenkompetenz Lernende erwerben die Fähigkeit selbstständig Testprogramme zum Bewerten der Leistungsfähigkeit eines Prozessors zu erstellen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Rechnerarchitektur Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Patterson/Hennessy: Computer Organization und Design • Hennessy/Patterson: Computer Architecture - A Quantitative Approach • Stallings: Computer Organization and Architecture • Märtin: Rechnerarchitekturen

Vertiefungsrichtung Software Engineering

1	**Modulbezeichnung** 97008	**Advanced Design and Programming (5-ECTS)** Advanced design and programming (5-ECTS)	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Dirk Riehle
5	**Inhalt**	This course teaches principles and practices of advanced object-oriented design and programming. Dieser Kurs wird auf Deutsch gehalten. It consists of a weekly lecture with exercises, homework and self-study. This is a hands-on course and students should be familiar with their Java IDE. Students learn the following concepts: Class-Level • Method design • Class design • Classes and interfaces • Subtyping and inheritance • Implementing inheritance • Design by contract Collaboration-Level • Values vs. objects • Role objects • Type objects • Object creation • Collaboration-based design • Design patterns Component-Level • Error handling • Meta-object protocols • Frameworks and components • Domain-driven design • API evolution The running example is the photo sharing and rating software Wahlzeit, see https://github.com/dirkriehle/wahlzeit . Class is held as a three hour session with a short break in between. Students should have a laptop ready with a working Java programming setup. Sign-up and further course information are available at https://adap.uni1.de - please sign up for the course on StudOn (available through previous link) as soon as possible. The course information will also tell you how the course will be held (online or in person).
6	**Lernziele und Kompetenzen**	• Students learn to recognize, analyze, and apply advanced concepts of object-oriented design and programming

		• Students learn to work effectively with a realistic tool set-up, involving an IDE, configuration management, and a service hoster
7	**Voraussetzungen für die Teilnahme**	INF-AuD or compatible / equivalent course
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Seminar Bachelor of Science Informatik 20092 Hauptseminar Bachelor of Science Informatik 20222 Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• See https://adap.uni1.de

1	**Modulbezeichnung** 510375	**Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML)** Analyzing and design object-oriented software systems with Unified Modeling Language (UML)	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Detlef Kips
5	**Inhalt**	Die sogenannte "Unified Modeling Language" (UML) ist der seit Jahren weltweit akzeptierte Notationsstandard für die Modellierung komplexer Softwaresysteme. Mit einem reichhaltigen Repertoire an graphischen und textuellen Ausdrucksmöglichkeiten bietet die UML ihren Anwendern die Möglichkeit, die Anforderungen an die Zielsoftware, ihre statischen bzw. dynamischen Systemeigenschaften sowie die gewählte Softwarearchitektur halbformal zu spezifizieren, im Team darüber zu kommunizieren und große Teile des Programmcodes aus den spezifizierten Systemmodellen zu generieren. Ziel dieser Vorlesung ist es, die Studierenden mit Syntax und Semantik der UML vertraut zu machen und zu demonstrieren, wie die UML im Rahmen eines "typischen" Softwareentwicklungsprozesses angewendet werden kann. Zu diesem Zweck werden die verschiedenen Diagrammtypen und Notationselemente der UML schrittweise eingeführt und anhand eines durchgängigen Anwendungsbeispiels im Rahmen eines konkreten Vorgehensmodells über alle Entwicklungsphasen hinweg eingesetzt.
6	**Lernziele und Kompetenzen**	Im Rahmen dieser Veranstaltung sollen die Studierenden insbesondere die Kompetenz erwerben, - die syntaktische Struktur und die Semantik vorgegebener UML-Modelle bzw. Modellausschnitte zu analysieren und zu erläutern - verschiedene Sprachelemente der UML (und ggf. deren Kombination) im Hinblick auf ihre Eignung zur Abbildung charakteristischer Modellierungsprobleme im Rahmen eines Softwareentwicklungsprozesses zu bewerten, auszuwählen und anzuwenden - die Struktur und Systematik des UML-Metamodells zu erläutern und die UML mit geeigneten Metamodellierungskonzepten auf spezifische Anwendungskontexte anzupassen - zu einer gegebenen Anforderungsdefinition im Rahmen einer systematischen Analyse- und Entwurfsmethodik ein integriertes UML-Systemmodell zu erstellen - den Funktionsumfang eines UML-basierten Modellierungswerkzeugs zu bewerten, ein geeignetes Werkzeug auszuwählen und sicher anzuwenden.
7	**Voraussetzungen für die Teilnahme**	Keine

8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Programmiersysteme Bachelor of Science Informatik 20222 Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich mündliche Einzelprüfung; Dauer (in Minuten): 30; benotet; 5 ECTS (Vorlesung + Übung)
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Rumbaugh, J.; Booch, G.; Jacobson, I.: The Unified Modeling Language Reference Manual, Addison-Wesley, 2004 • Hitz, M.; Kappel, G.; Kapsammer, E.; Retschitzegger, W.: UML @ work , 3., aktualisierte und überarbeitete Auflage, dpunkt-Verlag, 2005 • Winter, M.: Methodische objektorientierte Softwareentwicklung, dpunkt-Verlag, 2005 • Störrle, H.: UML 2 erfolgreich einsetzen, Addison-Wesley, 2007 • Rumpe, B.: Modellierung mit UML: Sprache, Konzepte und Methodik, Springer-Verlag, 2. Auflage, 2011 • Seidl, M., Brandsteidl, M., Huemer, C., Kappek, G.: UML@classroom - Eine Einführung in die objektorientierte Modellierung, dpunkt-Verlag, 2012 • Rupp, C.; Queins, S., et al. UML 2 glasklar: Praxiswissen für die UML-Modellierung, Carl Hanser Verlag, 2012 Die einschlägige Originalliteratur zur UML findet man auf der Website der Object Management Group (http://www.omg.org/spec/UML).

1	**Modulbezeichnung** 93002	**Datenbank Praxis** Database concepts in practice	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Florian Kronberger
5	**Inhalt**	**Inhalt** Datenbanken werden in fast jedem Unternehmen zur persistenten Datenspeicherung eingesetzt. Nach den Grundlagenvorlesungen im Bachelor, die die theoretische Einführung in die Datenbankwelt gegeben haben und die Basis für diesen Kurs bilden, wird in diesem Kurs die praktische Erfahrung in der Arbeit mit einem Datenbanksystem in den Fokus gerückt. Der grundlegende Theorie-Stoff wird mittels eines Online-Skripts als Selbstlernangebot angeboten. Daneben gibt es Vor-Ort Termine bei denen das gelernte Wissen teilweise wiederholt, vertieft und durch Praxisaufgaben gefestigt wird. Zudem kann in den Vor-Ort Terminen gezielt auf aufgetretene Probleme eingegangen und Fragen geklärt werden. Das in diesem Kurs verwendete Db2 for z/OS von IBM wird häufig im Enterprise-Umfeld eingesetzt. Insbesondere bei Banken, Versicherungsunternehmen und Softwarehäusern findet dieses Datenbanksystem Verwendung. Neben Oracle ist hier Db2 eines der weltweit am häufigsten eingesetzten Datenbanksysteme. Daneben wird im Kurs auch auf PostgreSQL als weiteren Vertreter der relationalen Datenbanksysteme eingegangen. Die Kursinhalte umfassen: • Wiederholung der grundlegenden Konzepte aus den Bachelor-Pflichtvorlesungen • Einführung und Überblick über Db2 for z/OS • Administration von Db2 for z/OS • Programmzugriff auf Db2 for z/OS • Tools für Db2 for z/OS • Angewandte Aufgaben anhand eines Praxisbeispiels • PostgreSQL
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Lernende erwerben Kenntnis der wesentlichen Begriffe aus dem Datenbank Umfeld, im Speziellen von Db2, sowie Kenntnisse über den Programmzugriff auf Datenbanken. Kenntnisse über die administrativen Aufgaben im Datenbank-Umfeld runden das Fachwissen der Studierenden ab. Verstehen Die Studierenden verstehen die Arbeitsweise des Datenbanksystems Db2. Sie könne Zugriffe auf das Datenbanksystem über Programme formulieren und verstehen den Mechanismus.

		Zusätzlich können sie administrative Tätigkeiten ausführen und verstehen und hinterfragen deren Wirkung. Anwenden Die Teilnehmer üben die Anwendung der Grundlagen aus dem Datenbankbereich im Umfeld der Db2, den Programmzugriff auf die Datenbank und Arbeiten mit SQL, zusätzlich wenden sie Administrationswerkzeuge auf die Db2-Datenbank an. Analysieren Am Schluss wird die Analyse und Konzeption einer Datenbank-Anwendung, sowie die Analyse von Datenbankzugriffen und Performanceproblemen, theoretisch und praktisch ducrhgeführt. Evaluieren (Beurteilen) Die Studierenden führen selbstständig die Evaluation einer bestehenden Datenbankarchitektur bezüglich der Effizienz und Einsetzbarkeit in einem gegebenen Kontext durch und müssen die Beurteilung von selbst erstellten Datenbankschemas und Datenbankzugriffsprogrammen erarbeiten.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Datenbanksysteme Bachelor of Science Informatik 20222 Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel (90 Minuten)
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Ist im StudOn-Kurs verlinkt

1	**Modulbezeichnung** 93550	**Grundlagen des Software Engineering** Foundations of software engineering	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Francesca Saglietti
5	**Inhalt**	Das Modul befasst sich mit einem breiten Spektrum an ingenieurwissenschaftlichen Prinzipien und alternativen Vorgehensweisen bei der Konzeption, Entwicklung, Analyse, Organisation und Pflege großer, komplexer Softwaresysteme.
6	**Lernziele und Kompetenzen**	Die Studierenden: • erklären systematische und strukturierte Prozessmodelle (wie das Wasserfall- und V-Modell) zur Entwicklung komplexer Software-Systeme • erfassen wesentliche Prinzipien der Software-Entwicklung (wie Kohäsion und Kopplung) • erläutern systematische Methoden zur Anforderungsermittlung an und wesentliche Ziele der Anforderungsanalyse • benutzen ausgewählte Spezifikationssprachen (wie Endliche Automaten, Petri-Netze und Z), um komplexe Problemstellungen eindeutig zu formulieren • wenden UML-Diagramme (wie Use Case-, Klassen-, Sequenz- und Kommunikationsdiagramme) zum Zweck objektorientierter Analyse- und Design-Aktivitäten an; • beschreiben unterschiedliche Arten der Wiederverwendung von Entwurfselementen und lösen typische Entwurfsprobleme durch Anwendung etablierter Entwurfsmuster • benutzen Software-Metriken zur Bestimmung der Software-Komplexität • wenden unterschiedliche Nachweisverfahren, wie z.B. statische Analyse, funktionales und strukturelles Testen, oder Korrektheitsbeweise an • stellen die Bedeutung der Wartung im Software-Entwicklungprozess heraus • erläutern verschiedene Aspekte des Software-Projektmanagements
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)

11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Lehrbuch der Softwaretechnik, Helmut Balzert, Spektrum Akad. Verlag, 2000

1	**Modulbezeichnung** 93183	**Mainframe@Home**	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	PD Dr. Peter Wilke
5	**Inhalt**	Großrechner sind das Herzstück der weltweiten IT-Landschaft. Durch die hohe Verfügbarkeit und geringe Ausfallquote werden Mainframes in sehr großen Firmen verwendet. Die Transaktionszahlen für die Datenverarbeitung sind bei diesen Unternehmen außerdem sehr hoch. Mit diesem Kurs soll Ihnen die Möglichkeit geboten werden, sich mit der Programmierung von Anwendungen für und der Arbeit mit Großrechner zu beschäftigen. Sie verwenden in diesem Kurs eine eigene Mainframe-Emulation auf Ihrem Rechner und arbeiten mit dieser in verschiedenen Übungsaufgaben. Behandelt werden die folgenden Kapitel: • Einführung in das Thema Großrechner • Virtualisierung • Multiple Virtual Storage (MVS) • Common Business Oriented Language (Cobol) • Formula Translator (Fortran) • Restructured Extended Executor (Rexx) • Virtual Storage Access Method (VSAM) • Java und Unix auf dem Mainframe
6	**Lernziele und Kompetenzen**	Wissen • Kenntnis der wesentlichen Begriffe der Mainframe-Arbeitsumgebung • Verständnis für das Arbeiten mit VSAM-Datasets Anwenden • Beherrschen der wichtigsten Kommandos zur Arbeit im Mainframe Betriebssystem MVS. • Aufbau einer eigenen Mainframe-Emulation mit MVS Betriebssystem. Verstehen • ° Analyse und Implementierung verschiedener Anwendung in den Sprache Cobol, Fortran und Rexx.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222

		Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 150 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Auf die Literatur wird in der jeweiligen Lerneinheit im StudOn hingewiesen.

1	**Modulbezeichnung** 580491	**Nailing your Thesis (PROJ 5-ECTS)** Nailing your thesis (PROJ 5-ECTS)	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Dirk Riehle
5	**Inhalt**	This course teaches students how to perform scientific research for their final thesis or a research paper. The goal is to prepare students for a Bachelor or Master research thesis. The course covers the following topics: • Science and society • The research process • Theory building research • Theory validation research • Writing a research thesis/paper • The scientific community Students can choose one or both of two components: • VUE (VL + UE or seminar), 4 SWS, 5 ECTS. VUE combines lectures with homework and exercises. VUE is run as a 3h block. • PROJ (small research project), 2 SWS, 5 ECTS. In PROJ, students perform a small research project, either individually or in teams. The available projects will be presented at the beginning of the course. Students perform the research, write a paper, and hold a presentation about their work. Sign-up and further course information are available at https://nyt.uni1.de - please sign up for the course on StudOn (available through previous link) as soon as possible. The course information will also tell you how the course will be held (online or in person).
6	**Lernziele und Kompetenzen**	• Students gain an understanding of how science works • Students learn how to perform research work • Students learn how to write a research thesis
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)

12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 0 h Eigenstudium: 150 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	See https://nyt.uni1.de

1	**Modulbezeichnung** 480491	**Nailing your Thesis (VUE 5-ECTS)** Nailing your thesis (VUE 5-ECTS)	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Dirk Riehle
5	**Inhalt**	This course teaches students how to perform scientific research for their final thesis or a research paper. The goal is to prepare students for a Bachelor or Master research thesis. The course covers the following topics: • Science and society • The research process • Theory building research • Theory validation research • Writing a research thesis/paper • The scientific community Students can choose one or both of two components: • VUE (VL + UE or seminar), 4 SWS, 5 ECTS. VUE combines lectures with homework and exercises. VUE is run as a 3h block. • PROJ (small research project), 2 SWS, 5 ECTS. In PROJ, students perform a small research project, either individually or in teams. The available projects will be presented at the beginning of the course. Students perform the research, write a paper, and hold a presentation about their work. Sign-up and further course information are available at https://nyt.uni1.de - please sign up for the course on StudOn (available through previous link) as soon as possible. The course information will also tell you how the course will be held (online or in person).
6	**Lernziele und Kompetenzen**	• Students gain an understanding of how science works • Students learn how to perform research work • Students learn how to write a research thesis
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Sommersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	See https://nyt.uni1.de

1	**Modulbezeichnung** 93145	**The AMOS Project (PO Role, VUE 5 ECTS)** The AMOS project (PO role)	**5 ECTS**
2	Lehrveranstaltungen	Übung: The AMOS Project (UE) (2 SWS) Vorlesung: The AMOS Project (VL) (2 SWS)	- -
3	Lehrende	Prof. Dr. Dirk Riehle	

4	**Modulverantwortliche/r**	Prof. Dr. Dirk Riehle
5	**Inhalt**	This course teaches agile methods (Scrum and XP) and open source tools using a single semester-long project. Topics covered are: • Agile methods and related software development processes • Scrum roles, process practices, including product and engineering management • Technical practices like refactoring, continuous integration, and test-driven development • Principles and best practices of open source software development The project is a software development project in which each student team works with an industry partner who provides the idea for the project. This is a practical hands-on experience. Students can play one of two primary roles: • Product owner. In this function, a student defines, prioritizes, communicates, and reviews requirements. The total effort adds up to 5 ECTS. • Software developer. In this function, a student estimates their effort for requirements and implements them. The total effort adds up to 10 ECTS. Students will be organized into teams of 7-8 people, combining product owners with software developers. An industry partner will provide requirements to be worked out in detail by the product owners and to be realized by the software developers. The available projects will be presented in the run-up to the course. Class consists of a 90min lecture followed by a 90min team meeting. Rooms and times for team meetings are assigned at the beginning of the semester. You must be able to regularly participate in the team meetings. If you can't, do not sign up for this course. Students choosing the software developer role must have prior software development experience. Sign-up and further course information are available at https://amos.uni1.de - please sign up for the course on StudOn (available through previous link) as soon as possible. The course information will also tell you how the course will be held (online or in person).
6	**Lernziele und Kompetenzen**	• Students learn about software products and software development in an industry context • Students learn about agile methods, in particular, Scrum and Extreme Programming

		• Students gain practical hands-on experience with a Scrum process and XP technical practices
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93143	**The AMOS Project (SD Role, VUE 10 ECTS)** The AMOS project (SD role)	**10 ECTS**
2	Lehrveranstaltungen	Übung: The AMOS Project (UE) (2 SWS) Vorlesung: The AMOS Project (VL) (2 SWS)	- -
3	Lehrende	Prof. Dr. Dirk Riehle	

4	**Modulverantwortliche/r**	Prof. Dr. Dirk Riehle
5	**Inhalt**	This course teaches agile methods (Scrum and XP) and open source tools using a single semester-long project. Topics covered are: • Agile methods and related software development processes • Scrum roles, process practices, including product and engineering management • Technical practices like refactoring, continuous integration, and test-driven development • Principles and best practices of open source software development The project is a software development project in which each student team works with an industry partner who provides the idea for the project. This is a practical hands-on experience. Students can play one of two primary roles: • Product owner. In this function, a student defines, prioritizes, communicates, and reviews requirements. The total effort adds up to 5 ECTS. • Software developer. In this function, a student estimates their effort for requirements and implements them. The total effort adds up to 10 ECTS. Students will be organized into teams of 7-8 people, combining product owners with software developers. An industry partner will provide requirements to be worked out in detail by the product owners and to be realized by the software developers. The available projects will be presented in the run-up to the course. Class consists of a 90min lecture followed by a 90min team meeting. Rooms and times for team meetings are assigned at the beginning of the semester. You must be able to regularly participate in the team meetings. If you can't, do not sign up for this course. Students choosing the software developer role must have prior software development experience. Sign-up and further course information are available at https://amos.uni1.de - please sign up for the course on StudOn (available through previous link) as soon as possible. The course information will also tell you how the course will be held (online or in person).
6	**Lernziele und Kompetenzen**	• Students learn about software products and software development in an industry context • Students learn about agile methods, in particular, Scrum and Extreme Programming

		• Students gain practical hands-on experience with a Scrum process and XP technical practices
7	**Voraussetzungen für die Teilnahme**	For software developer role: OSS-ADAP
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93184	**Commercial Open Source Startups (OSS-COSS)** Commercial open source startups (OSS-COSS)	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Dirk Riehle
5	**Inhalt**	This course teaches students how the software industry works, how commercial open source works, and how to spin-off from the university. It consists of four main components: • The software industry • Commercial open source • Open source projects • University spin-offs Class is run as two 90min blocks, one for the lecture, and one for the exercises. Sign-up and further course information are available at https://coss.uni1.de - please sign up for the course on StudOn (available through previous link) as soon as possible. The course information will also tell you how the course will be held (online or in person).
6	**Lernziele und Kompetenzen**	• Students learn about the software industry and its players • Students learn about commercial open source startups • Students learn how to get an open source project off the ground • Students learn how to spin-off a startup from university
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222 Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	See https://coss.uni1.de

1	**Modulbezeichnung** 57025	**Praktische Softwaretechnik** Applied software engineering	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Praktische Softwaretechnik (4 SWS)	5 ECTS
3	Lehrende	Ralf Ellner Dr.Ing. Christoph Erhardt	

4	**Modulverantwortliche/r**	Prof. Dr. Bernd Hindel Prof. Dr. Detlef Kips Prof. Dr. Dirk Riehle
5	**Inhalt**	Software ist überall und Software ist komplex. Nicht triviale Software wird von Teams entwickelt. Oft müssen bei der Entwicklung von Softwaresystemen eine Vielzahl von funktionalen und nicht funktionalen Anforderungen berücksichtigt werden. Hierfür ist eine disziplinierte und ingenieurmäßige Vorgehensweise notwendig. Die Vorlesung "Praktische Softwaretechnik" soll ... • ein Bewusstsein für die typischen Problemstellungen schaffen, die bei der Durchführung umfangreicher Softwareentwicklungsprojekte auftreten, • ein breites Basiswissen über die Konzepte, Methoden, Notationen und Werkzeuge der modernen Softwaretechnik vermitteln und • die Möglichkeiten und Grenzen ihres Einsatzes im Kontext realistischer Projektumgebungen anhand praktischer Beispiele demonstrieren und bewerten. Die Vorlesung adressiert inhaltlich alle wesentlichen Bereiche der Softwaretechnik. Vorgestellt werden unter anderem • traditionelle sowie agile Methoden der Softwareentwicklung, • Methoden der Anforderungsanalyse und des Systementwurfs, • Konzepte der Softwarearchitektur, -implementierung und Dokumentation und • Testen und Qualitätssicherung sowie Prozessverbesserung. Weitere Materialien und Informationen sind hier zu finden: • Zeitplan: http://goo.gl/0fy1T • Materialien: Auf StudOn über den Zeitplan Die Teilnahme ist begrenzt. Bitte registrieren Sie sich zeitig für den Kurs auf StudOn, um sicherzustellen, dass Sie einen Platz erhalten.
6	**Lernziele und Kompetenzen**	Die Studierenden • verstehen den Unterschied zwischen "Programmieren im Kleinen" und "Programmieren im Großen" (Softwaretechnik) • wenden grundlegende Methoden der Softwaretechnik über den gesamten Projekt- und Produktlebenszyklus an • kennen die Rolle und Zuständigkeiten der Berufsbilder "Projektleiter", "Anforderungsermittler", "Softwareentwickler" und "Qualitätssicherer"
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4

9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	siehe http://goo.gl/JSoUbV

1	**Modulbezeichnung** 93198	**Product Management** Product management	**10 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Praktikum Informatik Bachelor of Science Informatik 20222 Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 312443	**Software Projektmanagement** Software project management	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Software-Projektmanagement (4 SWS)	5 ECTS
3	Lehrende	Prof. Dr. Bernd Hindel	

4	**Modulverantwortliche/r**	Prof. Dr. Bernd Hindel
5	**Inhalt**	Zahlreiche Statistiken zeigen: Nur wenige Software-Projekte werden erfolgreich (hinsichtlich Zeit-, Budget- und Funktionsvorgaben) abgeschlossen. Sehr viele Projekte werden nur mit erheblichen Defiziten zu Ende gebracht, noch viel zu viele scheitern gänzlich. Oft liegen die Gründe im ungenügenden Projektmanagement. Die Vorlesung gibt einen Überblick zu grundlegenden Disziplinen des Projektmanagements und zeigt deren Wirkungsweisen an Hand von Praxisbeispielen. Gliederung: 1.Einführung Grundbegriffe des Projektmanagements, unterschiedliche Projektgrößen, unterschiedliche Projektarten, Erfolg und Misserfolg in Projekten 2.Projektstart und Planung, Kickoff-Meeting, Anforderungssammlung, Projektstrukturplan, Aufwandsschätzung, Aktivitäten-, Ressourcen- und Kostenplan 3.Projektkontrolle und Steuerung, Fortschrittsüberwachung, Besprechungen, Berichte, Änderungsmanagement 4.Personalmanagement, Der Faktor Mensch, Teamwork, Führungsgrundsätze, Gesprächsstrategien, Konflikte lösen 5.Änderungsmanagement Konfigurationen, Änderungswünsche, Change Control Board, Built- und Release-Mechanismen 6.Qualitäts- und Risikomanagement Qualitätsplan, Audits und Reviews, Risikoermittlung, Risikobewertung und Verfolgung, Gegenmaßnahmen 7.Reifegrad Modelle und Standards CMMI, SPiCE, ISO9001, ISO/IEC12207
6	**Lernziele und Kompetenzen**	Die Studierenden • kennen die Grundbegriffe des Projektmanagements • unterscheiden unterschiedliche Projektgrößen, unterschiedliche Projektarten • verstehen die Ursachen für Erfolg und Misserfolg in Projekten • planen selbständig Projekte und organisieren das Kickoff-Meeting • erstellen Anforderungen, Projektstrukturplan, Aufwandsschätzung, Aktivitäten-, Ressourcen- und Kostenplan • verstehen Projektkontrolle und Steuerung, Fortschrittsüberwachung, Besprechungen, Berichte, Änderungsmanagement

		• kennen die Grundzüge des Personalmanagements (Der Faktor Mensch, Teamwork, Führungsgrundsätze, Gesprächsstrategien, Konflikte lösen) • planen und steuern Änderungsmanagement (Konfigurationen, Änderungswünsche, Change Control Board, Built- und Release-Mechanismen) • setzen Qualitäts- und Risikomanagement ein (Qualitätsplan, Audits und Reviews, Risikoermittlung, Risikobewertung und Verfolgung, Gegenmaßnahmen) • kennen die wichtigsten Reifegrad Modelle und Standards (CMMI, SPiCE, ISO9001, ISO/IEC12207)
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	- Hindel, Bernd; Hörmann, Klaus; Müller, Markus; Schmied, Jürgen: "Basiswissen Software-Projektmanagement" (dpunkt-Verlag, 2. Auflage 2006) - Hindel, Bernd; Hörmann, Klaus; Müller, Markus; Dittmann, Lars: "SPiCE in der Praxis" (dpunkt-Verlag, 2006) - Hindel, Bernd; Versteegen, Gerhard; Meier, Erich; Vlasan, Adriana: "Prozessübergreifendes Projektmanagement" (Springer Verlag, 2005)

1	**Modulbezeichnung** 189989	**Testen von Softwaresystemen** Testing software systems	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Dr.-Ing. Norbert Oster
5	**Inhalt**	• Test-Terminologie und Software-Qualität nach ISO/IEC 9126 • Fundamentaler Testprozess • Teststufen im Softwarelebenszyklus • Statischer Test: Reviews • Erfahrungsbasiertes Testen • Black-Box-Testverfahren: Äquivalenzklassen-/Grenzwerttest, Zustandsbezogener Test, Entscheidungstabellentest • Statische Analyse: Daten- und Kontrollflussanomalien • White-Box-Testverfahren: Kontrollflussbasiert, Datenflussbasiert, Bedingungsüberdeckung • Mutationstest • Testmanagement: Planung, Kostenschätzung, Überwachung, Risikobewertung, Priorisierung, Fehlermanagement • Formale Verifikation: Theorem Proving und Model Checking • Quantitative Zuverlässigkeitsbewertung
6	**Lernziele und Kompetenzen**	Die Studierenden • unterscheiden die verschiedenen Fachbegriffe aus der Test-Domäne • erläutern die unterschiedlichen Aspekte der Software-Qualität • beschreiben den fundamentalen Testprozess und seine einzelnen Teilaufgaben • erläutern die wichtigsten Aspekte der Testpsychologie und entscheiden auf dieser Basis z.B. über das Testteam • beschreiben die typischen Teststufen und ordnen diese den Phasen im Softwarelebenszyklus zu • stellen die Unterschiede zwischen dynamischem Test, Review und statischer Analyse heraus • unterscheiden verschiedene Review-Arten und veranschaulichen deren typische Arbeitsschritte und Rollen • differenzieren unterschiedliche Formen erfahrungsbasierten Testens • wenden das Verfahren der Äquivalenzklassenbildung an und ermitteln entsprechende Testfälle für den Grenzwerttest • entwickeln Entscheidungstabellen für beliebige Testaufgaben und bestimmen die entsprechenden Testfälle • erläutern typische Daten-/Kontrollflussanomalien an selbst-gewählten Beispielen • unterscheiden verschiedene kontrollfluss-, datenfluss und bedingungsorientierte Testkriterien • wenden die grundlegenden White-Box-Testkriterien an und leiten entsprechende Testfälle ab

		• begründen die Ordnung der White-Box-Überdeckungskriterien in ihrer Subsumptionshierarchie • erläutern das Konzept des Mutationstests zur quantitativen Bewertung der Testgüte • gestalten einen konkreten Testprozess aus der Sicht eines Testmanagers • beschreiben die wichtigsten Facetten des Fehlermanagements • erläutern den Unterschied zwischen Theorem Proving und Model Checking und skizzieren das jeweilige Vorgehen • wenden Theorem Proving auf sequentiellen Code an und skizzieren den Beweis der Interferenzfreiheit bei Nebenläufigkeit • beschreiben Voraussetzungen, Annahmen und Vorgehen bei verschiedenen Arten der quantitativen Zuverlässigkeitsbewertung • entscheiden je nach Art des Softwareprodukts welche Art der quantitativen Zuverlässigkeitsbewertung zulässig ist • nennen die wichtigsten Normen und Standards sowie deren typische Anforderungen • bewerten und benutzen Werkzeuge für verschiedene Testaufgaben
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Programmiersysteme Bachelor of Science Informatik 20222 Vertiefungsrichtung Software Engineering Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Spillner, Andreas; Linz, Tilo: Basiswissen Softwaretest, dpunkt-Verlag • Liggesmeyer, Peter: Software-Qualität, Spektrum Verlag • Spillner, Andreas; Roßner, Thomas; Winter, Mario; Linz, Tilo: Praxiswissen Softwaretest - Testmanagement, dpunkt-Verlag • Lyu, Michael R.: Handbook of Software Reliability Engineering, McGraw-Hill

Vertiefungsrichtung Systemsimulation

1	**Modulbezeichnung** 465562	**Advanced Programming Techniques** Advanced programming techniques (lecture and exercises)	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Advanced Programming Techniques (4 SWS) Übung: ExAdvPT (2 SWS)	5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr. Harald Köstler Richard Angersbach	

4	**Modulverantwortliche/r**	Prof. Dr. Harald Köstler
5	**Inhalt**	Der Inhalt der Vorlesung besteht aus zahlreichen fortgeschrittenen C++-Themen, die ausgerichtet sind auf die richtige und effiziente Nutzung von C++ für eine professionelle Softwareentwicklung. The content of the lecture will consist of various topics of advanced C++ programming, aimed at teaching the proper and efficient usage of C++ for professional software development. These are basic language concepts, the newer standards (starting from C++11), object oriented programming in C++, static and dynamic polymorphism, template metaprogramming, and C++ idioms and design patterns. A good preparation for the lecture is the C++ primer book from S. Lippman et al. One should at least have several hundred hours of programming experience in C/C++ or any related object oriented programming language. Knowledge of basic concepts like pointers, references, inheritance and polymorphism is required.
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Lernende können die grundlegenden Sprachkonstrukte in den verschiedenen C++ Standards wiedergeben. Students know the basic language constructs from different C++ standards. Verstehen Lernende verstehen das C++ Objektmodell und können es mit anderen Programmiersprachen vergleichen. Students understand the C++ object model and are able to compare it to other programming languages. Anwenden Lernenden können Standardalgorithmen in einer objektorientierten Programmiersprache implementieren. Students can implement standard algorithms in an object oriented programming language. Analysieren Lernende können gängige Design Patterns klassifizieren und deren Anwendbarkeit für bestimmte Probleme diskutieren. Students are able to classify common design patterns and to discuss their usability for certain problems. Evaluieren (Beurteilen)

		Lernende können entscheiden, welches Software Design passend für eine bestimmte Aufgabe ist. Sie können auch den Implementierungsaufwand dafür abschätzen. Students can decide, which software design fits for a certain task. They are also able to estimate the programming effort for it. Erschaffen Lernende entwickeln selbständig in einer Gruppe ein größeres Softwarepaket im Bereich Simulation und Optimierung. Students develop together in a group a larger software project in the area of simulation and optimization on their own.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Systemsimulation Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio (60 Minuten)
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 165 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• S. Lippman: C++ Primer, Addison-Wesley • S. Meyers: Effective C++ Third Edition, Addison-Wesley • H. Sutter: Exceptional C++, Addison-Wesley

1	**Modulbezeichnung** 43740	**Programmiertechniken für Supercomputer** Programming techniques for supercomputers	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Gerhard Wellein
5	**Inhalt**	• Introduction to the architecture of modern supercomputers • Single core architecture and optimisation strategies • Memory hierarchy and data access optimization • Concepts of parallel computers and parallel computing • Efficient "shared memory parallelisation (OpenMP) • Parallelisation approaches for multi-core processors including GPUs • Efficient "distributed memory parallelisation (MPI) • Roofline performance model • Serial and parallel performance modelling
6	**Lernziele und Kompetenzen**	Students • acquire a comprehensive overview of programming modern supercomputers efficiently for numerical simulations • learn modern optimisation and parallelisation strategies, guided by structured performance modelling • acquire an insight into innovative programming techniques and alternative supercomputer architectures • are able to implement basic numerical methods with high hardware efficiency on parallel computers
7	**Voraussetzungen für die Teilnahme**	Programmierkenntnisse in C/C++ oder Fortran
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Systemsimulation Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• G. Hager and G. Wellein: Introduction to High Performance Computing for Scientists and Engineers. CRC Computational Science Series, 2010. ISBN 978-1439811924

		• J. Hennessy and D. Patterson: Computer Architecture. A Quantitative Approach. Morgan Kaufmann Publishers, Elsevier, 2003. ISBN 1-55860-724-2

1	**Modulbezeichnung** 278169	**Programming Techniques for Supercomputers (Lecture and Tutorial)** Programming techniques for supercomputers (lecture and tutorial)	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Gerhard Wellein
5	**Inhalt**	• Introduction to the architecture of modern supercomputers • Single core architecture and optimisation strategies • Memory hierarchy and data access optimization • Concepts of parallel computers and parallel computing • Efficient "shared memory parallelisation (OpenMP) • Parallelisation approaches for multi-core processors including GPUs • Efficient "distributed memory parallelisation (MPI) • Roofline performance model • Serial and parallel performance modelling • Energy efficient implementation and execution of parallel programs
6	**Lernziele und Kompetenzen**	Students • acquire a comprehensive overview of programming modern supercomputers efficiently for numerical simulations • learn modern optimisation and parallelisation strategies, guided by structured performance modelling • acquire an insight into innovative programming techniques and alternative supercomputer architectures • are able to implement basic numerical methods with high hardware efficiency on parallel computers • are able to program and use modern supercomputer with high (energy) efficiency
7	**Voraussetzungen für die Teilnahme**	Experience in C/C++ or Fortran programming required; basic knowledge of MPI and OpenMP programming
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Systemsimulation Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	G. Hager and G. Wellein: Introduction to High Performance Computing for Scientists and Engineers. CRC Computational Science Series, 2010. ISBN 978-1439811924 J. Hennessy and D. Patterson: Computer Architecture. A Quantitative Approach. Morgan Kaufmann Publishers, Elsevier, 2003. ISBN 1-55860-724-2

1	**Modulbezeichnung** 43370	**Simulation und Wissenschaftliches Rechnen 1** Simulation and scientific computing 1	**7,5 ECTS**
2	Lehrveranstaltungen	Tutorium: Tutorium zu Simulation und Wissenschaftliches Rechnen 1 (2 SWS) Übung: Tafelübung zu Simulation und Wissenschaftliches Rechnen 1 (2 SWS) Vorlesung: Vorlesung zu Simulation und Wissenschaftliches Rechnen 1 (2 SWS) Übung: Rechnerübung zu Simulation und wissenschaftliches Rechnen 1 (2 SWS)	2,5 ECTS 2,5 ECTS 2,5 ECTS -
3	Lehrende	Prof. Dr. Christoph Pflaum Dominik Thönnes	

4	**Modulverantwortliche/r**	Prof. Dr. Christoph Pflaum Prof. Dr. Ulrich Rüde
5	**Inhalt**	• Performance Optimierung für numerische Algorithmen • OpenMP Parallelisierung • Finite Differenzen Diskretisierung im Ort • Praktische Abschätzung des Diskretisierungsfehlers und der Konvergenzgeschwindigkeit numerischer Verfahren • Software Entwicklung im Bereich des wissenschaftlichen Rechnens • MPI Parallelisierung • Finite Differenzen Diskretisierung für zeitabhängige Probleme
6	**Lernziele und Kompetenzen**	Die Studierenden • lernen Techniken zur Optimierung von Algorithmen im Bereich des wissenschaftlichen Rechnens • lernen selbständig Algorithmen auf Parallelrechnern zu implementieren und zu optimieren • lernen theoretisch die Stabilität von numerischen Algorithmen zu untersuchen
7	**Voraussetzungen für die Teilnahme**	Voraussetzung ist ein Modul im Bereich Numerik
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Systemsimulation Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Deutsch oder Englisch
16	**Literaturhinweise**	• Lehrbuch: G. Hager und G. Wellein, Introduction to High Performance Computing for Scientists and Engineers, CRC Press, 2010. • Lehrbuch: Goedecker und Adolfy Hoisie. Performance Optimization of Numerically Intensive Codes, SIAM, 2001. • Lehrbuch: Gropp, Lusk, Skjellum, Using MPI. The MIT Press, 1999. • Lehrbuch: Alexandrescu, Modern C++ Design, Generic Programming and Design Patterns. Addison-Wesley, 2001. • Lehrbuch: Burden, Faires, Numerical Analysis, Brooks, 2001. • Lehrbuch: Chandra at. al., Programming in OpenMP, Academic Press, 2001.

Vertiefungsrichtung Theoretische Informatik

1	**Modulbezeichnung** 93169	**Advanced Mechanized Reasoning in Coq** Advanced mechanized reasoning in Coq	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	PD Dr.habil. Tadeusz Litak
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich (30 Minuten) Übungsleistung
11	**Berechnung der Modulnote**	mündlich (50%) Übungsleistung (50%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 56 h Eigenstudium: 169 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 861501	**Algebra des Programmierens** Algebra of programming (lecture with practical)	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Algebra des Programmierens (4 SWS)	7,5 ECTS
3	Lehrende	apl.Prof.Dr. Stefan Milius	

4	**Modulverantwortliche/r**	apl.Prof.Dr. Stefan Milius
5	**Inhalt**	Induktive Datentypen wie z.B. Listen, Stacks und Bäume werden abstrakt beschrieben. Strukturelle Induktion und Rekursion für solche Datentypen (z.B. die fold-Operation auf Listen) werden auf Grundlage der Initiale-Algebra-Semantik entwickelt. Dadurch werden verschiedene effektive Programmiertricks auf eine solide mathematische Grundlage gestellt. Grundlagen und Methoden der Kategorientheorie werden eingeführt und erklärt, insbesondere initiale Algebren und ihre Konstruktion. Evtl. werden Koalgebren behandelt, die es ermöglichen, verschiedene zustandsbasierte Systeme und ihre Semantik in einer einheitlichen Theorie zu studieren.
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben elementare Definitionen und Fakten zu induktiven Datentypen, zustandsbasierten Systemen und grundlegenden kategoriellen Begriffen wieder Verstehen Die Studierenden • erklären grundlegende Begriffe und Konzepte der Kategorientheorie • beschreiben Beispiele dieser Begriffe und Konzepte • erläutern grundlegende kategorielle Ergebnisse und deren Beweise Anwenden Die Studierenden • wenden strukturelle Induktionsprinzipien (Definitions- und Beweisprinzip) auf Programmierbeispiele an • wenden strukturelle Koinduktionsprinzipien (Definitions- und Beweisprinzip) auf festgelegte Typen von zustandbasierten Systemen (Koalgebren) an • übertragen kategorielle Begriffe auf verschiedene Spezialfälle der Informatik (abstrakte Datentypen und verschiedene Arten von Automaten) Analysieren Die Studierenden analysieren kategorielle Beweise, diskutieren die entsprechenden Argumentationen und legen diese schriftlich klar nieder. Erschaffen Die Studierenden

		• arbeiten strukturelle Induktions- und Koinduktionsprinzipien für neue Datentypen und zustandsbasierte Systeme als Spezialfall der Initiale-Algebra-Semantik und Terminale-Koalgebra-Semantik aus • kreieren kategorielle Begriffe, die Konzepte von konkreten Datentypen und Systemen abstrahieren Lern- bzw. Methodenkompetenz Die Studierenden • beherrschen das grundsätzliche Konzept des Beweises als hauptsächliche Methode des Erkenntnisgewinns in der theoretischen Informatik. Sie überblicken abstrakte Begriffsarchitekturen. • vollziehen mathematische Argumentationen nach, erklären diese, führen diese selbst und legen sie schriftlich nieder Sozialkompetenz Die Studierenden lösen Probleme in kollaborativer Gruppenarbeit und präsentieren erarbeitete Lösungen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 42 h Eigenstudium: 183 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	• R. Bird and O. de Moor: Algebra of Programming, Prentice Hall, 1996. • J. Adamek, H. Herrlich and G.E. Strecker: Abstract and Concrete Categories: The Joy of Cats, 2nd edition, Dover Publishers, 2009.

1	**Modulbezeichnung** 93079	**Algebraische und Logische Aspekte der Automatentheorie** Algebraic and logical aspects of automata theory	**5 ECTS**
2	Lehrveranstaltungen	Seminar: Algebraische und Logische Aspekte der Automatentheorie (2 SWS, WiSe 2023)	5 ECTS
3	Lehrende	apl.Prof.Dr. Stefan Milius	

4	**Modulverantwortliche/r**	apl.Prof.Dr. Stefan Milius
5	**Inhalt**	Automaten als mathematische Formalisierung zustandsbasierter Systeme gehören zu den wichtigsten Werkzeugen der Theoretischen Informatik und besitzen zahlreiche Anwendungen, von der Compilerentwicklung bis zur Verifikation reaktiver Systeme. In dieser Veranstaltung, die an die Anfängervorlesungen des Informatikstudiums anknüpft, werden wichtige Querverbindungen zwischen der Automatentheorie und Gebieten der Mathematik (Algebra, Topologie und Logik) hergestellt. Folgende Themen werden behandelt: • Erkennung von regulären Sprachen durch Monoide und Halbgruppen • Proendliche Gleichungen und Varietäten von Sprachen • Logische Beschreibung regulärer Sprachen, Ehrenfeucht-Fraïssé-Spiele • Automaten, Algebra und Logik auf unendlichen Wörtern und Bäumen
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben elementaren Definitionen, Begriffe und Fakten der algebraischen Sprachtheorie wieder. Verstehen Die Studierenden • erläutern grundlegende Begriffe der Halbgruppentheorie und der Topologie • erklären die Äquivalenz zwischen monadischer Logik 2. Stufe und regulären Sprachen • erklären die Äquivalenz zwischen Monoid erkennbaren und regulären Sprachen Anwenden Die Studierenden • beschreiben mit Hilfe von logischen Formeln formale Sprachen • konstruieren algebraische Erkenner für reguläre Sprachen • spezifizieren Klassen von regulären Sprachen durch Folgen von Gleichungen oder proendliche Gleichungen Analysieren Die Studierenden analysieren mathematische Beweise, diskutieren die entsprechenden Argumentationen und legen diese schriftlich nieder. Evaluieren (Beurteilen) Die Studierenden • ermitteln, ob eine vorgelegte Sprache durch die Logik erster Stufe beschreibbar ist Erschaffen Die Studierenden • entwickeln eine logische Sichtweise für weitere Automatenmodelle und Typen von formalen Sprachen. • arbeiten Korrespondenzen zwischen Automatenmodellen und algebraischen Strukturen aus. Lern- bzw. Methodenkompetenz Die Studierenden

		• beherrschen das grundsätzliche Konzept des Beweises als hauptsächliche Methode des Erkenntnisgewinns in der theoretischen Informatik. Sie überblicken abstrakte Begriffsarchitekturen. • vollziehen mathematische Argumentationen nach, erklären diese, führen diese selbst und legen sie schriftlich nieder. Sozialkompetenz Die Studierenden lösen Probleme in kollaborativer Gruppenarbeit und präsentieren erarbeitete Lösungen.
7	**Voraussetzungen für die Teilnahme**	Es wird empfohlen, folgende Module zu absolvieren, bevor dieses Modul belegt wird: • Grundlagen der Logik in der Informatik • Berechenbarkeit und formale Sprachen
8	**Einpassung in Studienverlaufsplan**	Semester: 1
9	**Verwendbarkeit des Moduls**	Hauptseminar Bachelor of Science Informatik 20222 Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Seminarleistung
11	**Berechnung der Modulnote**	Seminarleistung (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 45 h Eigenstudium: 180 h
14	**Dauer des Moduls**	1 Semester Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• J.-E. Pin: Mathematical Foundations of Automata Theory • D. Perrin, J.-E. Pin: Infinite Words, Academic Press, 2004 • H. Straubing: Finite Automata, Formal Logic, and Circuit Complexity, Birkhäuser, 1994 • E. Grädel, W. Thomas, T. Wilke: Automata, Logic, and Infinite Games, Springer, 2002

1	**Modulbezeichnung** 247639	**Approximationsalgorithmen** Approximation algorithms	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Rolf Wanka
5	**Inhalt**	Für viele kombinatorische Optimierungsprobleme hat sich herausgestellt, daß sie vermutlich nicht durch schnelle exakte Algorithmen gelöst werden können, weshalb man sich mit Näherungslösungen zufrieden geben muß. In dieser Vorlesung werden Approximationsalgorithmen vorgestellt, die für eine Reihe populärer Optimierungsprobleme beweisbar gute Lösungen in vertretbarer Zeit berechnen. Im ersten Teil der Veranstaltung werden die grundlegenden Begriffe vorgestellt, mit Beispielalgorithmen ausgeführt und jeweils die Grenzen aufgezeigt. Im zweiten Teil werden allgemeine Techniken eingeführt und anhand instruktiver Beispiele mit Leben erfüllt.
6	**Lernziele und Kompetenzen**	Die Studierenden lernen fortgeschrittene Konzepte für die approximative Lösung kombinatorischer Optimierungsproblem kennen und wie sie sie einsetzen können, um konkrete Anwendungsprobleme zu bearbeiten. Sie kennen dazu konkrete Einzelheiten wie Begriffe, Definitionen, Fakten, Gesetzmäßigkeiten und Theorien und lernen, wie die berechneten Lösungen analysiert und qualitativ mit der unbekannten optimalen Lösung in eine mathematisch Beziehung gesetzt werden.
7	**Voraussetzungen für die Teilnahme**	Die Module "Einführung in die Algorithmik" bzw. "Algorithmen und Datenstrukturen" und das Modul "Berechenbarkeit und Formale Sprachen".
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 165 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch

16	**Literaturhinweise**	- R. Wanka. Approximationsalgorithmen - Eine Einführung. Teubner, 2007. - K. Jansen, M. Margraf. Approximative Algorithmen und Nichtapproximierbarkeit. de Gruyter, 2008. - G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi. Complexity and Approximation -- Combinatorial Optimization Problems and Their Approximability Properties. Springer, 1999. - E. W. Mayr, H. J. Prömel, and A. Steger (Hrsg.). Lectures on Proof Verification and Approximation Algorithms. Springer, 1998. - V. V. Vazirani. Approximation Algorithms. Springer, 2001.

1	**Modulbezeichnung** 806144	**Beschreibungslogik und formale Ontologien** Description Logics and Formal Ontologies	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Lutz Schröder
5	**Inhalt**	• Algorithmen für Aussagenlogik • Tableaukalküle • Anfänge der (endlichen) Modelltheorie • Modal- und Beschreibungslogiken • Ontologieentwurf
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben Definitionen der Syntax und Semantik verschiedener WIssensrepräsentationssprachen wieder und legen wesentliche Eigenschaften hinsichtlich Entscheidbarkeit, Komplexität und Ausdrucksstärke dar. Anwenden Die Studierenden wenden Deduktionsalgorithmen auf Beispielformeln an. Sie stellen einfache Ontologien auf und führen anhand der diskutierten Techniken Beweise elementarer logischer Metaeigenschaften. Analysieren Die Studierenden klassifizieren Logiken nach grundlegenden Eigenschaften wie Ausdrucksstärke und Komplexität. Sie wählen für ein gegebenes Anwendungsproblem geeignete Formalismen aus. Lern- bzw. Methodenkompetenz Die Studierenden erarbeiten selbständig formale Beweise. Sozialkompetenz Die Studierenden arbeiten in Kleingruppen erfolgreich zusammen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4;5;6
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Künstliche Intelligenz Bachelor of Science Informatik 20222 Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 165 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	• M Krötzsch, F Simancik, I Horrocks; A description logic primer, arXiv, 2012 • F. Baader et al. (ed.): The Description Logic Handbook, Cambridge University Press, 2003 • M. Huth, M. Ryan: Logic in Computer Science, Cambridge University Press, 2004 • L. Libkin: Elements of Finite Model Theory, Springer, 2004

1	**Modulbezeichnung** 843472	**Effiziente kombinatorische Algorithmen** Efficient combinatorial algorithms	**7,5 ECTS**
2	Lehrveranstaltungen	Übung: UE-EffAlg (2 SWS) Vorlesung: Effiziente kombinatorische Algorithmen (2 SWS)	2,5 ECTS 5 ECTS
3	Lehrende	Prof. Dr. Rolf Wanka	

4	**Modulverantwortliche/r**	Prof. Dr. Rolf Wanka
5	**Inhalt**	In diesem Modul werden effiziente exakte Algorithmen für diskrete Probleme vorgestellt. Zuerst werden nichttriviale tiefensuchbasierte Linearzeitverfahren für die Berechnung zweifacher Zusammenhangskomponenten auf ungerichteten Graphen und starker Zusammenhangskomponenten auf gerichteten Graphen untersucht. Danach werden Polynomialzeit-Verfahren zur Berechnung maximaler Flüsse präsentiert. Eine Einführung in den Entwurf und die Analyse parametrisierter Algorithmen an Hand des Vertex-Cover-Problems und eine Einführung in den Bereich der sog. mild-exponentiellen Algorithmen für das Erfüllbarkeitsproblem und weiterer NP-vollständiger Probleme runden das Modul ab.
6	**Lernziele und Kompetenzen**	Die Studierenden lernen moderne fortgeschrittene Konzepte für die schnelle exakte Lösung kombinatorischer Optimierungsproblem kennen und wie sie sie einsetzen können, um konkrete Anwendungsprobleme zu bearbeiten. Sie kennen dazu konkrete fachspezifische Einzelheiten wie Begriffe, Definitionen, Fakten, Gesetzmäßigkeiten und Theorien und lernen und wie die berechneten Lösungen analysiertt und qualitativ eingeordnet werden.
7	**Voraussetzungen für die Teilnahme**	Die Module "Einführung in die Algorithmik" bzw. "Algorithmen und Datenstrukturen" und das Modul "Berechenbarkeit und Formale Sprachen".
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 165 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch

		• A. V. Aho, J. E. Hopcroft, J. D. Ullman. The Design and Analysis of Computer Algorithms. Addison-Wesley, 1975. • Venkatesh Raman, Saket Saurabh, Somnath Sikdar. Efficient Exact Algorithms through Enumerating Maximal Independent Sets and Other Techniques. Theory of Computing Systems 41 (2007) 563-587. • Frank Gurski, Irene Rothe, Jörg Rothe, Egon Wanke. Exakte Algorithmen für schwere Graphenprobleme. Springer 2010.
16	**Literaturhinweise**	• Sven Oliver Krumke, Hartmut Noltemeier. Graphentheoretische Konzepte und Algorithmen. Vieweg +Teubner, 2. Auflage 2009. • Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms (2nd Edition). MIT Press, 2001. • Fedor V. Fomin, Dieter Kratsch. Exact Exponential Algorithms. Springer, 2010. • Volker Heun. Grundlegende Algorithmen. Vieweg, 2. Auflage 2003. • Juraj Hromkovic. Algorithmics for Hard Problems. Springer, 2001. • Stephan Hußmann, Brigitte Lutz-Westphal (Hrsg.). Kombinatorische Optimierung erleben. Vieweg, 2007. • Jon Kleinberg, Eva Tardos. Algorithm Design. Pearson / Addison Wesley, 2006. • Sven Oliver Krumke, Hartmut Noltemeier. Graphentheoretische Konzepte und Algorithmen. Vieweg +Teubner, 2. Auflage 2009. • Christos H. Papadimitriou, Kenneth Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Dover Publications, 1998. • Volker Turau. Algorithmische Graphentheorie. Oldenbourg, 3. Auflage 2009. • Vöcking et al. (Hrsg.) Taschenbuch der Algorithmen. Springer 2008.

1	**Modulbezeichnung** 93540	**Ereignisgesteuerte Systeme**	**5 ECTS**
2	Lehrveranstaltungen	Übung: UE-EGS (2 SWS) Vorlesung: Ereignisgesteuerte Systeme (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Dr.-Ing. Stefan Wildermann	

4	**Modulverantwortliche/r**	Joachim Falk Dr.-Ing. Stefan Wildermann
5	**Inhalt**	Die rasante Entwicklung von Rechnertechnologien in den vergangenen Jahrzehnten hatte die Verbreitung neuer dynamischer und komplexer Systeme zu Folge. Wesentliche Charakteristika solcher Systeme sind Verteiltheit, Nebenläufigkeit und das asynchrone Auftreten diskreter Ereignisse. Der Prozess, neue Modelle und Methoden für ereignisgesteuerte Systeme zu entwickeln, ist vergleichsweise jung. Der Rechner selbst spielt hierbei eine entscheidende Rolle als Werkzeug für Systementwurf, Analyse und Steuerung. Das Modul EGS hat zum Ziel, Modellierungs-, Simulations- und Entwurfsmethoden für verteilte und ereignisdiskrete Systeme zu vermitteln. Die Methoden werden dabei beispielhaf auf Anwendung aus den Bereichen Computernetzwerke, automatischen Produktionssysteme, komplexen Softwaresysteme und integrierte Steuerungs-, Kommunikations- und Informationssysteme angewendet. In diesem Kontext behandelt das Modul daher die folgenden Themenbereiche: • Eigenschaften komplexer Systeme • Überblick über Systeme und Modelle • Zeitfreie und zeitbehaftete Modelle • Stochastische Modelle • Umsetzung in Programmiersprachen • Simulation-, Entwurfs- und Testverfahren auf der Basis der vorgestellten Modelle.
6	**Lernziele und Kompetenzen**	**Fachkompetenz - Verstehen** • Die Studierenden erläutern grundlegende Techniken zur Modellierung diskreter, ereignisgesteuerter Systeme, zeigen deren Vor- und Nachteile auf und vergleichen diese bezüglich Ihrer Mächtigkeit. **Fachkompetenz - Anwenden** • Die Studierenden wenden Modellierungs- und Analysetechniken aus dem Bereich endlicher Automaten, Petri-Netze, Markov-Ketten auf komplexe Systeme an. • Die Studierenden setzen die Modellierung und Analyse eines Systems mit einem konkreten Entwurfswerkzeug um.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4

9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222 Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/ereignisgesteuerte-systeme

1	**Modulbezeichnung** 151316	**Formale Methoden der Softwareentwicklung** Formal methods of software development	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	PD Dr.habil. Tadeusz Litak
5	**Inhalt**	In the first part of the course, we will engage in the formal verification of reactive systems. Students learn the syntax and semantics of the temporal logics LTL, CTL, and CTL** and their application in the specification of e.g. safety and liveness properties of systems. Simple models of systems are designed and verified using model checkers and dedicated frameworks for asynchronous and synchronous reactive systems, and the algorithms working in the background are explained. The second part of the course focuses on functional correctness of programs; more precisely, we discuss the theory of pre- and postconditions, Hoare triples, loop invariants, and weakest (liberal) preconditions, in order to introduce automatised correctness proofs using the Hoare calculus.
6	**Lernziele und Kompetenzen**	Students are going to acquire the following competences: Fachkompetenz Wissen • Reproduce the definition of syntax and semantics of temporal logics LTL, CTL, and CTL**. • Reproduce the definition of semantics of a simple programming languages like IMP, with special focus on axiomatic semantics (Hoare rules). • Explain how CTL can be characterised in terms of fixpoints. Verstehen The students understand the workings of state of the art automatic frameworks, clarifying the role of model checking algorithms, semantics and Hoare calculi in formal verification. Anwenden In a series of exercises, the students use state of the art tools for • model checking • specification and verification of reactive systems, • verification of functional correctness or memory safety of simple programs. Analysieren • Choose the optimal tool for a given verification or specification problem. • Differentiate between safety and liveness properties.

		• Contrast several related temporal logics (LTL, CTL, CTL**) and properties expressible/inexpressible in each of them.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 56 h Eigenstudium: 169 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	• G. Winskel: The Formal Semantics of Programming Languages: An Introduction, The MIT Press, 1993. • M. Huth, M. Ryan: Logic in Computer Science, Cambridge University Press, 2. Aufl., 2004.

1	**Modulbezeichnung** 173107	**Kommunikation und Parallele Prozesse** Communication and parallel processes	**7,5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Lutz Schröder
5	**Inhalt**	• Beschriftete Transitionssysteme • Prozessalgebren • Starke und schwache Bisimulation • Das Linear-Time/Branching-Time-Spektrum • Partition Refinment • Hennessy-Milner-Logik • Modaler mu-Kalkül
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben elementare Definitionen und Fakten zu reaktiven Systemen wieder. Verstehen Die Studierenden • erläutern semantische Grundbegriffe, insbesondere Systemtypen und Systemäquivalenzen, und identifizieren ihre wesentlichen Eigenschaften • erläutern die Syntax und Semantik von Logiken und Prozesskalkülen • fassen wesentliche Metaeigenschaften von Logiken und Prozesskalkülen zusammen. Anwenden Die Studierenden • übersetzen Prozessalgebraische Terme in ihre denotationelle und operationelle Semantik • prüfen Systeme auf verschiedene Formen von Bsimilarität • prüfen Erfüllheit modaler Fixpunktformeln in gegebenen Systemen • implementieren nebenläufige Probleme in Prozessalgebren • spezifizieren das Verhalten nebenläufiger Prozesse im modalen mu-Kalkül. Analysieren Die Studierenden • leiten einfache Meta-Eigenschaften von Kalkülen her • wählen für die Lösung gegebener nebenläufiger Probleme geeignete Formalismen aus Evaluieren (Beurteilen) Die Studierenden

		• vergleichen prozessalgebraische und logische Kalküle hinsichtlich Ausdrucksmächtigkeit und Berechenbarkeitseigenschaften • hinterfragen die Eignung eines Kalküls zur Lösung einer gegebenen Problemstellung Lern- bzw. Methodenkompetenz Die Studierenden beherrschen das grundsätzliche Konzept des Beweises als hauptsächliche Methode des Erkenntnisgewinns in der theoretischen Informatik. Sie überblicken abstrakte Begriffsarchitekturen. Sozialkompetenz Die Studierenden lösen abstrakte Probleme in kollaborativer Gruppenarbeit.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4;5;6
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	• Robin Milner, Communication and Concurrency, Prentice-Hall, 1989 • Julian Bradfield and Colin Stirling, Modal mu-calculi. In: Patrick Blackburn, Johan van Benthem and Frank Wolter (eds.), The Handbook of Modal Logic, pp. 721-756. Elsevier, 2006. • Jan Bergstra, Alban Ponse and Scott Smolka (eds.), Handbook of Process Algebra, Elsevier, 2006. • L. Aceto, A. Ingolfsdottir, K. Larsen and J. Srba, Reactive Systems, Cambridge University Press, 2011

1	**Modulbezeichnung** 984981	**Modallogik** Modal logic	**7,5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Lutz Schröder
5	**Inhalt**	• Relationale Modallogiken • Ausdrücksstärke • Aximoatisierung und Vollständigkeit • Entscheidbarkeit und Komplexität • Programmverifikation mittels dynamischer und temporaler Logiken • Modaler mu-Kalkül • Koalgebraische Logik
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben grundlegede Definitionen, Axiomatisierungen und Resultate aus der Metatheorie der Modallogik wieder. Anwenden Die Studierenden führen beispielhaft Beweise in modallogischen Deduktionssystemen und wenden modallogische Schlussfolgerungsalgorithmen korrekt an. Sie instanziieren gängige Vollständigkeitskritierien auf konkrete Modallogiken. Analysieren Die Studierenden teilen Logiken gemäß ihrer metalogischen Eigenschaften wie Axiomatisierbarkeit, Entscheidbarkeit oder Komplexität ein; sie wählen für gegebene Anwendungszwecke geeignete Logiken aus. Erschaffen Die Studierenden führen selbständig metatheoretische Beweise über Modallogiken. Lern- bzw. Methodenkompetenz Die Studierenden erarbeiten selbständig formale Beweise. Sozialkompetenz Die Studierenden arbeiten in Kleingruppen erfolgreich zusammen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4;5;6
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich

11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 56 h Eigenstudium: 169 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Sally Popkorn, |First Steps in Modal Logic|, 314 pages, Cambridge University Press, 1994. Patrick Blackburn, Maarten de Rijke and Yde Venema, |Modal Logic|, 554 pages, Cambridge University Press, 2001. Alexander Chagrov and Michael Zakharyashev, |Modal Logic|, 605 pages, Oxford University Press, 1997.

1	**Modulbezeichnung** 845618	**Monad-Based Programming** Monad-based programming	**7,5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	PD Dr.Ing. Sergey Goncharov
5	**Inhalt**	Der Kurs bietet einen Hintergrund zu verschiedenen Themen der Funktionale Programmierung. Als Leitparadigma wird die monadenbasierte funktionale Programmierung gewählt. Die Idee des Kurses ist es, klare berechnungsbezogene Einblicke in verschiedene Konzepte der Semantik und Programmierung zu geben und diese durch konkrete Implementierungen in der Programmiersprache Haskell zu üben. Zu diesem Zweck werden im Kurs die Sprache und der prinzipielle Ansatz der Kategorientheorie umfassend motiviert und verwendet. Schlüsselwörter: Monaden, Funktionale Programmierung, Kategorientheorie, Haskell, Equational reasoning; Kursseite: https://www8.cs.fau.de/monad-based-programming/
6	**Lernziele und Kompetenzen**	# Fachkompetenz ## Wissen Die Studierenden zeigen Verständnis der Rolle von Monaden im Kontext der funktionalen Programmierung und als semantisches Werkzeug für die Programmierung und Softwarespezifikation. Die Studierenden geben die wichtigsten Definitionen und Ergebnisse über Monaden, Monadenkombinationen und anderen kategorischen Konstrukten, sowie Produkte, Koprodukte, Funktoren, applikative Funktoren, exponentielle Objekte, wieder und erklären sie aus der Perspektive der Programmierung. ## Anwenden Die Studierenden verwenden den monadenbasierten Ansatz, um Beispiele mit verschiedenen Arten von Recheneffekten als Monaden zu formalisieren. Die Studierenden verwenden Monaden für die praktische Programmierung in Programmiersprachen, insbesondere in Haskell. ## Analysieren Die Studierenden identifizieren verschiedene Berechnungseffekte als Monaden und behandeln Probleme aus verschiedenen semantischen Domänen (zustandsabhängig, nichtdeterministisch, Ausnahmeverhalten) in geeigneter Weise und erstellen eine monadenbasierte Softwareimplementierung.

		Selbstkompetenz Die Studenten werden regelmäßig mit kleinen Herausforderungen in Form von Übungen versorgt, um einen allmählichen Fortschritt mit dem Vorlesungsmaterial zu erzielen.
7	**Voraussetzungen für die Teilnahme**	Als empfohlene Voraussetzung ist "Grundlagen der Logik in der Informatik" ausreichend. Alternativ sind allgemeine mathematische und logische Grundkenntnisse, die Sie auf jegliche Art und Weise erworben haben, ebenfalls geeignet. Stellen Sie sicher, dass Sie die folgenden Beispielfragen sicher beantworten können, um zu beurteilen, ob der Kurs für Sie geeignet ist (eine richtige Antwort muss nicht eindeutig sein, aber wenn Sie die Frage im Wesentlichen verstehen, sollte es kein Problem sein): - Wann sind zwei Funktionen, sagen wir f und g, von einer Menge A zu einer Menge B gleich? - Wann sind zwei Mengen gleich? - Wann eine Relation zwischen zwei Mengen A und B eine Funktion ist? - Wenn eine Implikation A -> B falsch ist, muss dann auch A falsch sein? Muss B falsch sein? - Verstehen Sie das Prinzip des "Beweis durch Widerspruch"? - Was ist das (Kartesische) Produkt von zwei Mengen? - Was ist eine disjunkte Vereinigung von zwei Mengen? - Was ist das Prinzip der vollständigen Induktion? - Erläutern Sie den Zusammenhang zwischen endlichen binären Bäumen und Termen über einer finitären Signatur. Welche Aritäten die beteiligten Operationen haben werden? - Was sind Boolesche Konstanten? - Wie lautet die Definition der Fakultätsfunktion? Wie lautet die Definition der Fibonacci-Zahlen? - Wie lautet die Formel zur Berechnung der Summe der vollen Quadrate $1^2 + ... + n^2$? - Was ist die Definition einer teilweise geordneten Menge? - Was ist eine Äquivalenz?
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30, benotet, 7.5 ECTS Anteil an der Berechnung der Modulnote: 100.0% weitere Erläuterungen: Die Modulnote setzt sich zu 50% aus dem Ergebnis einer 30-minütigen mündlichen Prüfung am Semesterende und zu

		50% aus der Note für die Bearbeitung von 6 Übungsaufgabenblättern zusammen. Prüfungssprache: Englisch oder Deutsch
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 165 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 599478	**Praktische Semantik von Programmiersprachen** Practical semantics of programming languages	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	PD Dr.habil. Tadeusz Litak
5	**Inhalt**	We study the foundations of the imperative and functional languages, including semantics and type systems. The special feature of this course is that theory is done in a very practical and hands-on way: we not just prove, but program all the results from first-principles. The basic tool used in the course is Coq proof assistant, which can be regarded as a functional programming language in its own right. It has been used, for example, to verify correctness of Java Card technology, C compilers or, more recently, fragments of x86 architecture.
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen The students explain the basics of both programming semantics and proof assistants, in particular Coq. Verstehen The students prove theorems using a proof assistant. Anwenden The students transfer proofs into programs and programs into proofs. Analysieren The students examine behaviour of simple programs using formal semantics Evaluieren (Beurteilen) The students evaluate the role played by logic and type theory in scientific approach to programming. Erschaffen The students provide formal semantics to a simple programming language.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 56 h Eigenstudium: 169 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Online book "Software Foundations" http://www.cis.upenn.edu/~bcpierce/sf/ Online books by Adam Chlipala: "Certified Programming with Dependent Types" http://adam.chlipala.net/cpdt/ and "Formal Reasoning About Programs" http://adam.chlipala.net/frap/ Supplementary reading on the theory of programming: Types and Programming Languages Benjamin C. Pierce, The MIT Press Supplementary reading on Coq: Interactive Theorem Proving and Program Development Coq'Art: The Calculus of Inductive Constructions Series: Texts in Theoretical Computer Science. An EATCS Series Bertot, Yves, Casteran, Pierre

1	**Modulbezeichnung** 164985	**Randomisierte Algorithmen** Randomised algorithms	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Rolf Wanka
5	**Inhalt**	Bei der Lösung kombinatorischer oder zahlentheoretischer Probleme ist es oft möglich, durch Würfeln schnell und einfach mit hoher Wahrscheinklichkeit oder im Durchschnitt zu hervorragenden Lösungen zu kommen. In diesem Modul lernen wir Konzepte wie die Probabilistische Methode, Irrläufe (Random Walks) und Varianzanalysen von Zufallsprozessen kennen und wenden sie auf graphentheoretische Probleme und effiziente Datenstrukturen an. Zu den vorgestellten Inhalten gehören u.a.: • Schnelle Wiederholung wahrscheinlichkeitstheoretischer Begriffe und Resultate • Das Pólyasche Urnen-Modell und Chernoff-Schranken • Die Probabilistische Methode und ihre Anwendung auf die Berechnung maximaler Schnitte und unabhängiger Mengen und die Anwendung der Probabilistischen Methode zum Beweis der Lovász-Local-Lemma • Random Walks und ihre Anwendung auf das Erfüllbarkeitsproblem • Approximate Counting und die Markov-Chain-Monte-Carlo-Methode Neueste Ergebnisse dieses Forschungsgebietes werden inhaltlich in das Modul eingebunden.
6	**Lernziele und Kompetenzen**	Die Studierenden lernen moderne fortgeschrittene Konzepte für die schnelle Lösung kombinatorischer Optimierungsproblem mithilfe zufallsbasierter Algorithmen kennen und wie sie sie einsetzen können, um konkrete Anwendungsprobleme zu bearbeiten. Sie kennen dazu konkrete fachspezifische Einzelheiten wie Begriffe, Definitionen, Fakten, Gesetzmäßigkeiten und Theorien und lernen und wie die berechneten Lösungen analysiertt und qualitativ eingeordnet werden.
7	**Voraussetzungen für die Teilnahme**	Die Module "Einführung in die Algorithmik" bzw. "Algorithmen und Datenstrukturen" und das Modul "Berechenbarkeit und Formale Sprachen".
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)

12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 165 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	• Michael Mitzenmacher, Eli Upfal: Probability and Computing - Randomization and Probabilistic Techniques in Algorithms and Data Analysis (2nd ed.). Cambridge University Press, 2017 • Juraj Hromkovic. Randomisierte Algorithmen. Teubner, 2004. • Rajeev Motwani, Prabhakar Raghavan: Randomized Algorithms. Cambridge University Press, 1995.

1	**Modulbezeichnung** 44500	**Swarm Intelligence**	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Rolf Wanka
5	**Inhalt**	Swarm Intelligence (SI) is the design and deployment of self-organizing systems that dynamically adapt to their respective environmental needs. These systems are characterized by the fact that they feature the so-called self-*-properties, i.e., they are self-configuring, self-optimizing, self-healing, self-protecting, self-explanatory, ... Structures and methods of biological and other natural systems are chosen as models for such technical systems. In this module, Particle Swarm Optimization, Ant Algorithms, Web Search, and Evolutionary Algorithms are introduced and, as far as possible, mathematically analyzed.
6	**Lernziele und Kompetenzen**	Students learn advanced concepts of the current topic of swarm intelligence and how they can be successfully applied to solve continuous and discrete optimization problems and to data analysis. For this purpose, they know concrete details such as terms, definitions, facts, regularities and theories and learn how to apply the concepts to concrete problems, how to adjust the methods to the use case and how to analyze the computed solutions.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222 Vertiefungsrichtung Theoretische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch

16	**Literaturhinweise**	• Ch. Müller-Schloer, Ch. von der Malsburg, R. P. Würt. Organic Computing. Informatik-Spektrum, Band 27, Nummer 4, S. 332-336. • I. C. Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters 85 (2003) 317-325. • J. M. Kleinberg. Authoritative sources in a hyperlinked environment. Journal of the ACM 46 (1999) 604-632. • M. Dorigo. V. Maniezzo. A Colorni. Ant system: an autocatalytic optimizing process. Technical Report 91-016, Politecnico di Milano, 1991. • A. Badr. A. Fahmy. A proof of convergence for Ant algorithms. Information Sciences 160 (2004) 267-279. • M. Clerc. J. Kennedy. The particle swarm - Explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 8 (2002) 58-73

Vertiefungsrichtung Verteilte Systeme und Betriebssysteme

1	**Modulbezeichnung** 820947	**Betriebssysteme (Vorlesung mit erweiterten Übungen)** Operating systems (lectures with extend exercises)	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Betriebssysteme (2 SWS) Übung: BS-RÜ (0 SWS) Übung: BS-EÜ (4 SWS)	2,5 ECTS - 5 ECTS
3	Lehrende	Dr.-Ing. Volkmar Sieh Maximilian Ott Phillip Raffeck Dustin Nguyen Jonas Rabenstein	

4	**Modulverantwortliche/r**	Dr.-Ing. Volkmar Sieh
5	**Inhalt**	Inhalt des Moduls ist Vermittlung grundlegender Konzepte, Methoden und Techniken von Betriebssystemen aus dem Blickwinkel einer Betriebssystementwicklerin. Im Rahmen der Übungen entwickeln die Studierenden in einem "bottom-up" Entwurf- und Entwicklungsprozess ihr eigenes Mehrkernbetriebssystem für die IA-32 Platform, ausgehend von der "nackten Hardware" über grundlegende Ein-Ausgabemöglichleiten, Unterbrechungsbearbeitung bis hin zu quasi- und echtparalleler Programmausführung.
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: - erläutern den Startvorgang eines Rechensystems am Beispiel eines x86/x86_64-PCs. - beschreiben die spezifischen Herausforderungen bei der Softwareentwicklung für "bare metal". - beschreiben den Ablauf einer Unterbrechungsbehandlung von der Hardware bis zur (System-)software. - skizzieren Besonderheiten und Strategien der Unterbrechungsbehandlung in Hardware für Mehrkernsystemen am Beispiel des IA32-APICs. - diskutieren die Aufgabenteilung zwischen Hardware und Systemsoftware bei der Unterbrechungsbearbeitung. - unterscheiden die verschiedenen Typen von Kontrollflüssen in einem Betriebssystem anhand des Ebenenmodells. - unterscheiden harte, mehrstufige, und weiche Verfahren zur Unterbrechungssynchronisation in Betriebssystemen und können diese für ein Mehrkernsystem implementieren. - klassifizieren konkrete Konkurrenzsituationen anhand des Ebenenmodels und leiten daraus geeignete Synchronisationsmaßnahmen ab. - schildern die IA32-Architektur und gängige PC-Technologie und deren Schnittstellen zur Systemsoftware. - erläutern grundlegende Bausteine für die Implementierung von Quasi-Parallelität (Fortsetzungen, Koroutinen, Fäden) und grenzen diese gegeneinander ab. - erläutern die Interaktionen zwischen Hardware, Übersetzer und Systemsoftware, die dabei zu beachten sind.

		- entwickeln den Koroutinenwechsel für einen gegebene Architektur. - erläutern die Implikationen von Quasi-Parallität auf das Ebenenmodell und die daraus abgeleiteten Synchronisationsmaßnahmen. - beschreiben die Implementierung von (verdrängendem) Scheduling in einem Betriebssystem. - analysieren das Zusammenspiel von Scheduling und Unterbrechungssynchronisation. - nennen Kriterien und Dimensionen des Schedulings von Betriebsmitteln, insbesondere der CPU. - erläutern die konkrete Umsetzung am Beispiel der Scheduler in Linux und Windows. - unterscheiden grundlegende Möglichkeiten der Koordinierung und Synchronisation von Fäden (aktives/passives Warten, nichtverdrängbare kritische Abschnitte). - entwickeln Mechanismen für die Synchronisation auf Fadenebene für Mehrkernsystemen. - erklären die dabei zu beachtenden Synchronisationsprobleme (lost update, lost wakeup) und geeignete Gegenmaßnahmen. - interpretieren die Bedeutung von Gerätetreibern in der Betriebssystempraxis. - erläutern die Anforderungen an ein Treibermodell. - vergleichen die Umsetzung von Treibermodellen in Windows und Linux. - vergleichen grundlegende BS-Architekturen (Bibliothek, Monolith, Mikrokern, Exokern, Hypervisor) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und Mechanismen. - schildern die grundlegenden Paradigmen zur Interprozesskommunikation in Betriebssystemen (speicherbasiert vs. nachrichtenbasiert). - erläutern die grundlegenden Primitiven dieser Verfahren. - skizzieren, wie unter Anwendung dieser Primitiven höhere Synchronisationskonstrukte implementiert werden (Monitore, Leser-/Schreiber-Sperre). - illustrieren die Dualität der Paradigmen. - erschließen sich typische Probleme (Nebenläufigkeit, Compilerverhalten, Debuggen ohne dedizierte Hilfsmittel) und Fehlerquellen bei der hardwarenahen Softwareentwicklung für Mehrkernsysteme. - können in Kleingruppen kooperativ arbeiten. - können die ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. - reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab. - können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.
7	**Voraussetzungen für die Teilnahme**	• C/C++ (Übungsaufgaben werden in C++ implementiert) • Assembler (Grundkenntnisse)
8	**Einpassung in Studienverlaufsplan**	Semester: 5

9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Avi Silberschatz, Peter B. Galvin, Greg Gagne. Operating System Concepts. John Wiley & Sons. • Andrew S. Tanenbaum. Modern Operating Systems. Pearson. • William Stallings. Operating Systems: Internals and Design Principles. Prentice Hall. • Andrew S. Tanenbaum. Structured Computer Organization. Pearson.

1	**Modulbezeichnung** 150033	**Betriebssysteme (Vorlesung mit Übungen)** Operating systems (lecture with exercises)	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Betriebssysteme (2 SWS) Übung: BS-Ü (2 SWS) Übung: BS-RÜ (0 SWS)	2,5 ECTS 2,5 ECTS -
3	Lehrende	Dr.-Ing. Volkmar Sieh Dustin Nguyen Jonas Rabenstein Phillip Raffeck Maximilian Ott	

4	**Modulverantwortliche/r**	Dr.-Ing. Volkmar Sieh
5	**Inhalt**	Inhalt des Moduls ist Vermittlung grundlegender Konzepte, Methoden und Techniken von Betriebssystemen aus dem Blickwinkel einer Betriebssystementwicklerin. Im Rahmen der Übungen entwickeln die Studierenden in einem "bottom-up" Entwurf- und Entwicklungsprozess ihr eigenes Einkernbetriebssystem für die IA-32 Platform, ausgehend von der "nackten Hardware" über grundlegende Ein-Ausgabemöglichleiten, Unterbrechungsbearbeitung bis hin zu quasiparalleler Programmausführung.
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: - erläutern den Startvorgang eines Rechensystems am Beispiel eines x86/x86_64-PCs. - beschreiben die spezifischen Herausforderungen bei der Softwareentwicklung für "bare metal". - beschreiben den Ablauf einer Unterbrechungsbehandlung von der Hardware bis zur (System-)software. - skizzieren Besonderheiten und Strategien der Unterbrechungsbehandlung in Hardware für Mehrkernsystemen am Beispiel des IA32-APICs. - diskutieren die Aufgabenteilung zwischen Hardware und Systemsoftware bei der Unterbrechungsbearbeitung. - unterscheiden die verschiedenen Typen von Kontrollflüssen in einem Betriebssystem anhand des Ebenenmodells. - unterscheiden harte, mehrstufige, und weiche Verfahren zur Unterbrechungssynchronisation in Betriebssystemen und können diese implementieren. - klassifizieren konkrete Konkurrenzsituationen anhand des Ebenenmodels und leiten daraus geeignete Synchronisationsmaßnahmen ab. - schildern die IA32-Architektur und gängige PC-Technologie und deren Schnittstellen zur Systemsoftware. - erläutern grundlegende Bausteine für die Implementierung von Quasi-Parallelität (Fortsetzungen, Koroutinen, Fäden) und grenzen diese gegeneinander ab. - erläutern die Interaktionen zwischen Hardware, Übersetzer und Systemsoftware, die dabei zu beachten sind. - entwickeln den Koroutinenwechsel für einen gegebene Architektur.

		- erläutern die Implikationen von Quasi-Parallität auf das Ebenenmodell und die daraus abgeleiteten Synchronisationsmaßnahmen. - beschreiben die Implementierung von (verdrängendem) Scheduling in einem Betriebssystem. - analysieren das Zusammenspiel von Scheduling und Unterbrechungssynchronisation. - nennen Kriterien und Dimensionen des Schedulings von Betriebsmitteln, insbesondere der CPU. - erläutern die konkrete Umsetzung am Beispiel der Scheduler in Linux und Windows. - unterscheiden grundlegende Möglichkeiten der Koordinierung und Synchronisation von Fäden (aktives/passives Warten, nichtverdrängbare kritische Abschnitte). - entwickeln Mechanismen für die Synchronisation auf Fadenebene. - erklären die dabei zu beachtenden Synchronisationsprobleme (lost update, lost wakeup) und geeignete Gegenmaßnahmen. - interpretieren die Bedeutung von Gerätetreibern in der Betriebssystempraxis. - erläutern die Anforderungen an ein Treibermodell. - vergleichen die Umsetzung von Treibermodellen in Windows und Linux. - vergleichen grundlegende BS-Architekturen (Bibliothek, Monolith, Mikrokern, Exokern, Hypervisor) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und Mechanismen. - schildern die grundlegenden Paradigmen zur Interprozesskommunikation in Betriebssystemen (speicherbasiert vs. nachrichtenbasiert). - erläutern die grundlegenden Primitiven dieser Verfahren. - skizzieren, wie unter Anwendung dieser Primitiven höhere Synchronisationskonstrukte implementiert werden (Monitore, Leser-/Schreiber-Sperre). - illustrieren die Dualität der Paradigmen. - erschließen sich typische Probleme (Nebenläufigkeit, Compilerverhalten, Debuggen ohne dedizierte Hilfsmittel) und Fehlerquellen bei der hardwarenahen Softwareentwicklung. - können in Kleingruppen kooperativ arbeiten. - können die ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. - reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab. - können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.
7	**Voraussetzungen für die Teilnahme**	• C/C++ (Übungsaufgaben werden in C++ implementiert) • Assembler (Grundkenntnisse)
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Avi Silberschatz, Peter B. Galvin, Greg Gagne. Operating System Concepts. John Wiley & Sons. • Andrew S. Tanenbaum. Modern Operating Systems. Pearson. • William Stallings. Operating Systems: Internals and Design Principles. Prentice Hall. • Andrew S. Tanenbaum. Structured Computer Organization. Pearson.

1	**Modulbezeichnung** 350752	**Betriebssystemtechnik (V+EÜ)** Operating systems engineering	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Wolfgang Schröder-Preikschat Dr.-Ing. Volkmar Sieh
5	**Inhalt**	Schwerpunktthema der Veranstaltung ist die Verwaltung von Prozessadressräumen. Untersucht werden Verfahren und Techniken zur Trennung logischer Adressräume, zum Adressraumgrenzen überschreitenden Zugriff und zum Schutz von Prozessen. Vorgestellt wird die Implementierung von Systemaufrufen und seiten- wie auch segmentbasierte Techniken zur Abbildung logischer/ virtueller Adressräume auf reale. Vor diesem Hintergrund werden verschiedene Betriebssystemarchitekturen verglichen und gängige Adressraummodelle von Betriebssystemen erläutert. Weitere Themen bildet die Interprozesskommunikation durch Nachrichtenversenden bei getrennten Adressräumen, aber auch die Nachbildung virtuell gemeinsamen Speichers auf Basis solcher Ansätze.
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: - erläutern und implementieren Entwurfsprinzipien für Systemaufrufe und diskutieren deren spezifischen Vor-/Nachteile. - vergleichen grundlegende BS-Architekturen (Monolith, Mikrokern, Makrokern, Exokern) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und deren Einfluss auf die Implementierung von Mechanismen (Systemaufrufe, Addressraumschutz). - unterscheiden Hierarchiekonzepte des Softwareentwurfs (Benutzthierachie, funktionale Hierachie) und erläutern deren Implikationen beim Betriebssystementwurf. - klassifizieren Schutz-, Verwaltungs-, und Virtualisierungstechniken für Programmzustände (Seitennummerierung, Segementierung, Sprachbasierung, Capabilities) und implementieren diese auf der x86_64-Architektur - diskutieren Adressraummodelle (Mehradressraummodell, Einadressraummodell, mehrstufige und inverse Seitenabbildungen, Mitbenutzung) und deren Implementierbarkeit auf gängigen Hardwarearchitekturen. - implementieren Mechanismen und Abtraktionen zur Interprozesskommunikation. - erläutern das Zusammenspiel zwischen Kommunikation und Synchronisation im Bezug auf die besonderen Herausforderungen bei der Implementierung von Betriebssytemabstraktionen (lost wakeup, lost update, gepufferte/ungepufferte Zugriffe) - diskutieren Prinzipien der Mitbenutzung von Code und Daten unter Berücksichtigung der Betriebssystem- und Addressraumarchitektur.

		- erläutern die Funktionsweise eines Bindeladers und skizzieren effiziente Implementierungstechniken für positionsunabhängige Strukturen. - erschließen sich typische Probleme (Nebenläufigkeit, Compilerverhalten, Debuggen ohne dedizierte Hilfsmittel) und Fehlerquellen bei der hardwarenahen Softwareentwicklung. - können in Kleingruppen kooperativ arbeiten. - können die ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. - reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab. - können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.
7	**Voraussetzungen für die Teilnahme**	• C/C++ (Übungsaufgaben werden in C++ implementiert) • Assembler (Grundkenntnisse)
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Wolfgang Schröder-Preikschat. The Logical Design of Parallel Operating Systems. Prentice Hall. • Brian W. Kernighan, Dennis M. Ritchie. The C Programming Language. Prentice Hall. • David M. Weiss, Chi Tau Robert Lai. Software Product-Line Engineering: A Family-Based Software Development Process. Addison-Wesley. • Krysztof Czarnecki, Ulrich W. Eisenecker. Generative Programming. Methods, Tools and Applications. Addison-Wesley.

1	**Modulbezeichnung** 406841	**Betriebssystemtechnik (Vorlesung mit Übungen)** Operating system technology	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Wolfgang Schröder-Preikschat Dr.-Ing. Volkmar Sieh
5	**Inhalt**	Schwerpunktthema der Veranstaltung ist die Verwaltung von Prozessadressräumen. Untersucht werden Verfahren und Techniken zur Trennung logischer Adressräume, zum Adressraumgrenzen überschreitenden Zugriff und zum Schutz von Prozessen. Vorgestellt wird die Implementierung von Systemaufrufen und seiten- wie auch segmentbasierte Techniken zur Abbildung logischer/ virtueller Adressräume auf reale. Vor diesem Hintergrund werden verschiedene Betriebssystemarchitekturen verglichen und gängige Adressraummodelle von Betriebssystemen erläutert. Weitere Themen bildet die Interprozesskommunikation durch Nachrichtenversenden bei getrennten Adressräumen, aber auch die Nachbildung virtuell gemeinsamen Speichers auf Basis solcher Ansätze.
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: - erläutern und implementieren Entwurfsprinzipien für Systemaufrufe und diskutieren deren spezifischen Vor-/Nachteile. - vergleichen grundlegende BS-Architekturen (Monolith, Mikrokern, Makrokern, Exokern) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und deren Einfluss auf die Implementierung von Mechanismen (Systemaufrufe, Addressraumschutz). - unterscheiden Hierarchiekonzepte des Softwareentwurfs (Benutzthierachie, funktionale Hierachie) und erläutern deren Implikationen beim Betriebssystementwurf. - klassifizieren Schutz-, Verwaltungs-, und Virtualisierungstechniken für Programmzustände (Seitennummerierung, Segementierung, Sprachbasierung, Capabilities) und implementieren diese auf der x86_64-Architektur - diskutieren Adressraummodelle (Mehradressraummodell, Einadressraummodell, mehrstufige und inverse Seitenabbildungen, Mitbenutzung) und deren Implementierbarkeit auf gängigen Hardwarearchitekturen. - implementieren Mechanismen und Abtraktionen zur Interprozesskommunikation. - erläutern das Zusammenspiel zwischen Kommunikation und Synchronisation im Bezug auf die besonderen Herausforderungen bei der Implementierung von Betriebssytemabstraktionen (lost wakeup, lost update, gepufferte/ungepufferte Zugriffe) - diskutieren Prinzipien der Mitbenutzung von Code und Daten unter Berücksichtigung der Betriebssystem- und Addressraumarchitektur.

		- erläutern die Funktionsweise eines Bindeladers und skizzieren effiziente Implementierungstechniken für positionsunabhängige Strukturen. - erschließen sich typische Probleme (Nebenläufigkeit, Compilerverhalten, Debuggen ohne dedizierte Hilfsmittel) und Fehlerquellen bei der hardwarenahen Softwareentwicklung. - können in Kleingruppen kooperativ arbeiten. - können die ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. - reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab. - können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.
7	**Voraussetzungen für die Teilnahme**	• C/C++ (Übungsaufgaben werden in C++ implementiert) • Assembler (Grundkenntnisse)
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Wolfgang Schröder-Preikschat. The Logical Design of Parallel Operating Systems. Prentice Hall. • Brian W. Kernighan, Dennis M. Ritchie. The C Programming Language. Prentice Hall. • David M. Weiss, Chi Tau Robert Lai. Software Product-Line Engineering: A Family-Based Software Development Process. Addison-Wesley. • Krysztof Czarnecki, Ulrich W. Eisenecker. Generative Programming. Methods, Tools and Applications. Addison-Wesley.

1	**Modulbezeichnung** 939179	**Echtzeitsysteme 2 - Verlässliche Echtzeitsysteme** Real-time systems 2 - dependable real-time systems	**7,5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Peter Wägemann
5	**Inhalt**	Viele Echtzeitsysteme sind in Bereiche des täglichen Lebens eingebettete, die hohe Anforderungen an die funktionale Sicherheit dieser Systeme stellen. Beispiele hierfür sind Fahrerassistenzsysteme in modernen Automobilen, medizinische Geräte, Prozessanlagen in Kernkraftwerken oder Chemiefabriken oder Flugzeuge. Fehlfunktionen in diesen Anwendungen ziehen mitunter katastrophale Konsequenzen nach sich - Menschen können ernsthaft verletzt oder sogar getötet werden, Landstriche können unbewohnbar gemacht oder zumindest großer finanzieller Schaden verursacht werden. Dieses Modul betrachtet Methoden und Werkzeuge, die uns helfen können, einerseits *zuverlässig Software zu entwickeln* (also Fehler im Programm zu entdecken und zu vermeiden), und andererseits *zuverlässige Software zu entwickeln* (also Abstraktionen, die auch im Fehlerfall ihre Gültigkeit behalten). Hierbei steht weniger die Vermittlung theoretischer Grundkenntnisse auf diesen Gebieten im Vordergrund, also vielmehr • die praktische Anwendung existierende Werkzeuge und Methoden • sowie die Erfahrung und das Verständnis ihrer Grenzen. Auf diese Weise soll ein Fundament für die konstruktive Umsetzung verlässlicher Echtzeitsysteme gelegt werden. Dieses Modul soll daher fundierte Anknüpfungspunkte für die Entwicklung verlässlicher Echtzeitsysteme vermitteln, die Ad-hoc-Techniken möglichst ersetzen sollen.
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: • nennen die Konzepte und die Taxonomie verlässlicher Systeme, unterscheiden Software- und Hardwarefehler und klassifizieren Fehler (Defekt, Fehler, Fehlverhalten). • stellen Fehlerbäume auf. • organisieren Softwareentwicklungsprojekte mittels der Versionsverwaltung git. • vergleichen die verschiedenen Arten der Redundanz als Grundvorraussetzung für Fehlererkennung und -toleranz. • entwickeln fehlertolerante Systeme mittels Replikation. • diskutieren die Fehlerhypothese und die Sicherstellung von Replikdeterminismus. • erläutern die Vor- und Nachteile softwarebasierter Replikation und den Einsatz von Diversität.

		• wenden Informationsredundanz zur Härtung von Daten- und Kontrollflüssen an. • bewerten die Effektivität der arithmetischer Codierung von Programmen und verallgemeinern diesen Ansatz auf die verschiedenen Implementeirungsebenen (Maschinenprogramm zu Prozessinkarnation). • interpretieren den Einfluss der Ausführungsplattform (Hardware, Betriebssystem) auf die Leistungsfähigkeit der Fehlererkennung. • konzipieren eine fehlertolerante Ausführungsumgebung für ein softwarebasiertes TMR-System basierend auf ANBD-Codierung. • nennen die Grundlagen der systematischen Fehlerinjektion. • überprüfen die Wirksamkeit von Fehlertoleranzmechanismen mittels Fehlerinjektion auf der Befehlssatzebene. • entwickeln Testfälle für die Fehlerinjektion mittels des fail** Werkzeugs. • setzten Messergebnisse in Relation zu dem tatsächlichen Fehlerraum. • beschreiben die Grundlagen der Fehlererholung (Vorwärts- bzw. Rückwärtskorrektur) und Reintegration fehlgeschlagener Knoten. • vergleichen den Zustandstransfer am Beispiel der Running bzw. Recursive State Restoration. • benennen Konzepte der Rückwärtskorrektur durch Entwurfsalternativen (Recovery Blocks). • fassen die Grundlagen des dynamischen Testens zusammen. • unterscheiden Black-Box und White-Box Testverfahren. • konzipieren und implementieren Testfälle. • überprüfen die Testüberdeckung anhand grundlegender Überdeckungskriterien (Anweisungs- bis Bedingungsüberdeckung). • geben die Grundlagen der statischen Programmanalyse wieder. • nennen die Funktionsweise von Hoare- WP-Kalkül. • verifizieren eine Ampelsteuerung mittels des FramaC Werkzeugs zur statischen Analyse von C Programmen. • beschreiben den Korrektheitsnachweis mittels abstrakter Interpretation und unterscheiden die konkrete von der abstrakten Programmsemnatik. • erläutern die Funktionsweise von Sammel- und Präfixsemantiken. • erstellen einen Korrektheitsbeweis für einen a-b-Filter mittels des Astrée Werkzeugs zur abstrakten Interpretation von C Programmen. • bewerten die Verlässlichkeit kommerzieller, sicherheitskritischer Systeme anhand von Fallstudien (Sizewell B, Airbus A320).

		• erschließen sich typische Probleme und Fehlerquellen bei der Programmierung von eingebetteten Systemen im Allgemeinen. • klassifizieren Fallstricke und Mehrdeutigkeiten in der Programmiersprache C99 im Besonderen. • können in Gruppen kooperativ und effektiv arbeiten. • können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. • reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab. • können offen und konstruktiv mit Schwachpunkten in der Konzeption wie Umsetzung umgehen.
7	**Voraussetzungen für die Teilnahme**	Für eine erfolgreiche Teilnahme an der Veranstaltung sind grundlegende Programmierkenntnisse in C/C++ erforderlich. Diese können durch den (empfohlenen) Besuch entsprechender Grundlagenveranstaltungen oder im Eigenstudium erworben sein. Weiterhin sind grundlegende Kenntnisse über Echtzeitsystemeeine, zum Beispiel durch den Besuch der Veranstaltung "Echtzeitsysteme", empfohlen. Eine formale Voraussetzung besteht in diesem Zusammenhang jedoch nicht.
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 179490	**Echtzeitsysteme mit erweiterten Übungen** Real-time systems with extended exercises	**7,5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Peter Wägemann
5	**Inhalt**	Videobearbeitung in Echtzeit, Echtzeitstrategiespiel, echtzeitfähig - der Begriff Echtzeit ist wohl einer der am meisten strapazierten Begriffe der Informatik und wird in den verschiedensten Zusammenhängen benutzt. Diese Vorlesung beschäftigt sich mit dem Begriff Echtzeit aus der Sicht von Betriebssystemen - was versteht man eigentlich unter dem Begriff Echtzeit im Betriebssystemumfeld, wo und warum setzt man sog. Echtzeitbetriebssysteme ein und was zeichnet solche Echtzeitbetriebssysteme aus? In dieser Vorlesung geht es darum, die oben genannten Fragen zu beantworten, indem die grundlegenden Techniken und Mechanismen vermittelt werden, die man im Betriebssystemumfeld verwendet, um Echtzeitsysteme und Echtzeitbetriebssysteme zu realisieren. Im Rahmen dieser Vorlesung werden unter anderem folgende Themen behandelt: • zeitgesteuerte und ereignisgesteuerte Systeme • statische und dynamische Ablaufplanungsverfahren • Fadensynchronisation in Echtzeitbetriebssystemen • Behandlung von periodischen und nicht-periodischen Ereignissen In den begleitenden Übungen werden die in der Vorlesung vorgestellten Techniken bei der Entwicklung eines kleinen Echtzeitsystems praktisch umgesetzt.
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: • unterscheiden die verschiedenen Komponenten eines Echtzeitsystems. • bewerten die Verbindlichkeiten von Terminvorgaben (weich, fest, hart). • erläutern die Zusammensetzung des Laufzeitverhaltes einer Echtzeitanwendung. • klassifizieren die Berührungspunkte zwischen physikalischem Objekt und kontrollierendem Echtzeitsystem. • interpretieren die Zeitparameter des durch das Echtzeitrechensystem zu kontrollierenden Objekts. • nennen die Zeitparameter des zugrundeliegenden Rechensystems (Unterbrechungslatenz, Ausführungszeit, ...). • unterscheiden synchrone und asynchrone Programmunterbrechung (insbesondere Trap/Interrupt, Ausnahmebehandlung und Zustandssicherung).

		• skizzieren die Verwaltungsgemeinkosten des schlimmsten Falls. • entwickeln in der Programmiersprache C und wenden die GNU Werkzeugkette für den ARM Cortex M4 an. • erstellen Echtzeitanwendungen auf Basis der eCos OS-Schnittstelle • ordnen die Strukturelemente von Echtzeitanwendungen zu: Aufgabe, Arbeitsauftrag und Faden. • erläutern die Implikationen von zeitlichem Mehrfachbetrieb auf die Verwaltungsgemeinkosten. • unterscheiden die Umsetzungsalternativen zur Ablaufsteuerung und die Trennung der Belange in Einplanung (Strategie) und Einlastung (Mechanismus). • benennen die grundsätzliche Verfahren der Ablaufsteuerung (taktgesteuert, reihum, vorranggesteuert). • erklären die grundlegenden Zeitparameter einer Aufgabe (Auslösezeitpunkt, Termin, Antwortzeit, Latenz, Ausführungszeit, Schlupfzeit). • unterscheiden die Grundlagen der Planbarkeit (gültig vs. zulässig, Optimalität von Einplanungsalgorithmen). • beschreiben den Unterschied zwischen konstruktiver und analytischer Einhaltung von Terminen-. • vergleich die Möglichkeiten (statisch, dynamisch) der zeitliche Analyse von Echtzeitanwendungen. • erklären die Grundlagen und Beschränkungen von dynamischer (worst-case?) und statischer WCET-Analyse (makroskopisch und mikroskopisch). • illustrieren Lösungsverfahren zur Bestimmung des längsten Ausführungspfads (Timing Schema, IPET). • erstellen Zeitmessung mittels Zeitgeber / Oszilloskop und bestimmen den längsten Pfad durch Code-Review. • erproben und konzipieren werkzeuggestützte WCET-Analyse mittels des absint aiT Analysewerkzeugs. • Entwickeln und annotieren Flußrestrikionen für die statische WCET-Analyse. • beschreiben die Grundlagen der Abfertigung periodischer Echtzeitsysteme (Periode, Phase, Hyperperiode). • skizzieren das periodische Modell und dessen Folgen (Entwicklungskomfort vs. Analysierbarkeit). • erklären die ereignisgesteuerte Ausführung (feste und dynamische Priorität, Verdrängbarkeit) mittels ereignisorienterter Planer (Berechnungskomplexität, MLQ-Scheduler, O(1)-Scheduler). • unterscheiden die zeitgesteuerte Ausführung (Busy Loop, Ablaufplan)und die Abfertigung von Arbeitsaufträgen im Abfrage- bzw. Unterbrecherbetrieb. • wenden die Grundlagen der ereignisgesteuerten Ablaufplanung periodischer Echtzeitsysteme an.

		• unterscheiden Verfahren zur statischen (RM, DM) und dynamischen Prioritätsvergabe (EDF, LRT, LST). • nennen den Unterschied zwischen Anwendungs- und Systemebene (Mehrdeutigkeit von Prioritäten). • erläutern den Optimalitätsnachweis des RM-, DM- und EDF-Algorithmus und dessen Ausnahmen. • beschreiben grundlegende Verfahren zur Planbarkeitsanalyse (CPU-Auslastung, Antwortzeitanalyse). • implementieren komplexe Aufgabensysteme in eCos. • unterscheiden die Grundlagen der zeitgesteuerten Ablaufplanung periodischer Echtzeitsysteme. • erstellen regelmäßige, zyklische Ablaufpläne (cyclic executive model, Rahmen). • vergleich Methoden der manuellen und algorithmischen Ablaufplanung. • unterscheiden optimale von heuristischen Verfahren (List Scheduling, Branch & Bound). • diskutieren die Konsequenzen eines Betriebswechsels in Echtzeitsystemen. • erstellen takt- beziehungsweise ereignisgesteuerte Abläufe in eCos beziehungsweise tt-eCos. • entwickeln ein softwarebasiertes Oszilloskop und erstellen dessen zeitliche Analyse und Ablaufplanung. • klassifizieren die Grundlagen der Abfertigung nicht-periodischer Echtzeitsysteme (minimale Zwischenankunftszeit). • definieren die Verbindlichkeiten von nicht-periodischen Aufgaben (aperiodisch, sporadisch) • zeigen die sich ergebenden Restriktionen des periodischen Modells (Mischbetrieb, Prioritätswarteschlangen, Übernahmeprüfung) auf. • beschreiben die Basistechniken des Laufzeitsystems (Zusteller, Unterbrecherbetrieb, Hintergrundbetrieb). • quantifizieren die Eigenschaften und Auswirkungen auf den periodischen Teil des Echtzeitsystems. • formulieren die Grundlagen des Slack-Stealing. • beschreiben den Einsatz von bandweite-bewahrenden Zustellern. • unterscheiden aufschiebbare Zusteller und Sporadic Server (SpSL und POSIX). • wenden eine Übernahmeprüfung bei sporadischen Aufgaben mittels dichte- oder schlupfbasierten Akzeptanztests an. • arbeiten einen strukturierter Ablaufplan (Rahmen) aus und untersuchen den Einsatz von Slack-Stealing. • ermitteln gerichtete Abhängigkeiten und Rangfolgen in Echtzeitanwendungen (Abhängigkeits- und Aufgabengraph). • stellen Umsetzungsalternativen für Abhängigkeiten einander gegenüber (naiv, implizit, explizit).

		• beschreiben das Konzept der zeitlichen Domänen und physikalischer bzw. logischer Ereignisse. • übertragen Abhängigkeiten auf das Problem der Ablaufplanung (modifiziere Auslösezeitpunkt/Termin, Phasenversatz). • konzipieren Rangfolge und aperiodische Steuerung in eCos. • implementieren einen aperiodischer Moduswechsel mit Zustandsüberführung in eCos. • gestalten einen Signal-Trigger für das entwickelte softwarebasierten Oszilloskops. • konzipieren explizite Synchronisation mittels Nachrichten in eCos. • wenden die Grundlagen von Wettstreit um Betriebsmitteln, Konkurrenz und Konfliktsituationen (kritische Abschnitte, (un)kontrollierte Prioritätsumkehr) an. • beschreiben echtzeitfähige Synchronisationsprotokolle (NPCS, PI, PCP). • nennen die Vor- und Nachteile der Techniken (transitive Blockung, Verklemmungen). • hinterfragen die Vereinfachung des PCP durch stapelbezogene Grenzprioritäten. • bestimmen die Ablaufplanung unter Berücksichtigung von Blockierungszeiten und Selbstsuspendierung. • implementieren Zugriffskontrolle (NPCS, PI, PCP) in Echtzeitanwendungen mit eCos. • analysieren Blockade für die Zugriffskontrolle in eCos. • erläutern die Anforderungen an verteile Echtzeitsysteme (Komposition, Erweiterbarkeit, Komplexität, Ereignis- vs. Zustandsnachricht). • fassen die Grundlagen von Knoten, Netzwerkschnittstellen und Netzübergängen sowie die Konzepte der expliziten und impliziten Flusskontrolle zusammen. • erschließen sich typische Probleme (zeitliche Analyse, Beobachtbarkeit, Synchronisation, Rangfolge) und Fehlerquellen bei der Programmierung von Echtzeitanwendungen. • können in Gruppen kooperativ und effektiv arbeiten. • können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. • reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab. • können offen und konstruktiv mit Schwachpunkten in der Konzeption wie Umsetzung umgehen.
7	**Voraussetzungen für die Teilnahme**	Für eine erfolgreiche Teilnahme an der Veranstaltung sind grundlegende Programmierkenntnisse in C/C++ erforderlich. Diese können durch den (empfohlenen) Besuch entsprechender Grundlagenveranstaltungen oder im Eigenstudium erworben sein, eine formale Voraussetzung besteht in diesem Zusammenhang nicht.

8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Hermann Kopetz. Real-Time Systems: Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers, 1997. • Jane W. S. Liu. Real-Time Systems. Prentice-Hall, Inc., 2000. • Wolfgang Schröder-Preikschat. System-programmierung. Vorlesungsfolien. 2006.

1	**Modulbezeichnung** 707303	**Echtzeitsysteme (Vorlesung mit Übungen)** Real-time systems	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Peter Wägemann
5	**Inhalt**	Videobearbeitung in Echtzeit, Echtzeitstrategiespiel, echtzeitfähig - der Begriff Echtzeit ist wohl einer der am meisten strapazierten Begriffe der Informatik und wird in den verschiedensten Zusammenhängen benutzt. Diese Vorlesung beschäftigt sich mit dem Begriff Echtzeit aus der Sicht von Betriebssystemen - was versteht man eigentlich unter dem Begriff Echtzeit im Betriebssystemumfeld, wo und warum setzt man sog. Echtzeitbetriebssysteme ein und was zeichnet solche Echtzeitbetriebssysteme aus? In dieser Vorlesung geht es darum, die oben genannten Fragen zu beantworten, indem die grundlegenden Techniken und Mechanismen vermittelt werden, die man im Betriebssystemumfeld verwendet, um Echtzeitsysteme und Echtzeitbetriebssysteme zu realisieren. Im Rahmen dieser Vorlesung werden unter anderem folgende Themen behandelt: • zeitgesteuerte und ereignisgesteuerte Systeme • statische und dynamische Ablaufplanungsverfahren • Fadensynchronisation in Echtzeitbetriebssystemen • Behandlung von periodischen und nicht-periodischen Ereignissen In den begleitenden Übungen werden die in der Vorlesung vorgestellten Techniken bei der Entwicklung eines kleinen Echtzeitsystems praktisch umgesetzt.
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: • unterscheiden die verschiedenen Komponenten eines Echtzeitsystems. • bewerten die Verbindlichkeiten von Terminvorgaben (weich, fest, hart). • erläutern die Zusammensetzung des Laufzeitverhaltes einer Echtzeitanwendung. • klassifizieren die Berührungspunkte zwischen physikalischem Objekt und kontrollierendem Echtzeitsystem. • interpretieren die Zeitparameter des durch das Echtzeitrechensystem zu kontrollierenden Objekts. • nennen die Zeitparameter des zugrundeliegenden Rechensystems (Unterbrechungslatenz, Ausführungszeit, ...). • unterscheiden synchrone und asynchrone Programmunterbrechung (insbesondere Trap/Interrupt, Ausnahmebehandlung und Zustandssicherung).

- skizzieren die Verwaltungsgemeinkosten des schlimmsten Falls.
- entwickeln in der Programmiersprache C und wenden die GNU Werkzeugkette für den ARM Cortex M4 Microcontroller an.
- erstellen Echtzeitanwendungen auf Basis der eCos OS-Schnittstelle
- ordnen die Strukturelemente von Echtzeitanwendungen zu: Aufgabe, Arbeitsauftrag und Faden.
- erläutern die Implikationen von zeitlichem Mehrfachbetrieb auf die Verwaltungsgemeinkosten.
- unterscheiden die Umsetzungsalternativen zur Ablaufsteuerung und die Trennung der Belange in Einplanung (Strategie) und Einlastung (Mechanismus).
- benennen die grundsätzliche Verfahren der Ablaufsteuerung (taktgesteuert, reihum, vorranggesteuert).
- erklären die grundlegenden Zeitparameter einer Aufgabe (Auslösezeitpunkt, Termin, Antwortzeit, Latenz, Ausführungszeit, Schlupfzeit).
- unterscheiden die Grundlagen der Planbarkeit (gültig vs. zulässig, Optimalität von Einplanungsalgorithmen).
- beschreiben den Unterschied zwischen konstruktiver und analytischer Einhaltung von Terminen-.
- vergleich die Möglichkeiten (statisch, dynamisch) der zeitliche Analyse von Echtzeitanwendungen.
- erklären die Grundlagen und Beschränkungen von dynamischer (worst-case?) und statischer WCET-Analyse (makroskopisch und mikroskopisch).
- illustrieren Lösungsverfahren zur Bestimmung des längsten Ausführungspfads (Timing Schema, IPET).
- erstellen Zeitmessung mittels Zeitgeber / Oszilloskop und bestimmen den längsten Pfad durch Code-Review.
- erproben werkzeuggestützte WCET-Analyse mittels des absint aiT Analysewerkzeugs.
- beschreiben die Grundlagen der Abfertigung periodischer Echtzeitsysteme (Periode, Phase, Hyperperiode).
- skizzieren das periodische Modell und dessen Folgen (Entwicklungskomfort vs. Analysierbarkeit).
- erklären die ereignisgesteuerte Ausführung (feste und dynamische Priorität, Verdrängbarkeit) mittels ereignisorienterter Planer (Berechnungskomplexität, MLQ-Scheduler, O(1)-Scheduler).
- unterscheiden die zeitgesteuerte Ausführung (Busy Loop, Ablaufplan)und die Abfertigung von Arbeitsaufträgen im Abfrage- bzw. Unterbrecherbetrieb.
- wenden die Grundlagen der ereignisgesteuerten Ablaufplanung periodischer Echtzeitsysteme an.
- unterscheiden Verfahren zur statischen (RM, DM) und dynamischen Prioritätsvergabe (EDF, LRT, LST).

- nennen den Unterschied zwischen Anwendungs- und Systemebene (Mehrdeutigkeit von Prioritäten).
- erläutern den Optimalitätsnachweis des RM-, DM- und EDF-Algorithmus und dessen Ausnahmen.
- beschreiben grundlegende Verfahren zur Planbarkeitsanalyse (CPU-Auslastung, Antwortzeitanalyse).
- implementieren komplexe Aufgabensysteme in eCos.
- unterscheiden die Grundlagen der zeitgesteuerten Ablaufplanung periodischer Echtzeitsysteme.
- erstellen regelmäßige, zyklische Ablaufpläne (cyclic executive model, Rahmen).
- vergleich Methoden der manuellen und algorithmischen Ablaufplanung.
- unterscheiden optimale von heuristischen Verfahren (List Scheduling, Branch & Bound).
- diskutieren die Konsequenzen eines Betriebswechsels in Echtzeitsystemen.
- erstellen takt- beziehungsweise ereignisgesteuerte Abläufe in eCos beziehungsweise tt-eCos.
- klassifizieren die Grundlagen der Abfertigung nicht-periodischer Echtzeitsysteme (minimale Zwischenankunftszeit).
- definieren die Verbindlichkeiten von nicht-periodischen Aufgaben (aperiodisch, sporadisch)
- zeigen die sich ergebenden Restriktionen des periodischen Modells (Mischbetrieb, Prioritätswarteschlangen, Übernahmeprüfung) auf.
- beschreiben die Basistechniken des Laufzeitsystems (Zusteller, Unterbrecherbetrieb, Hintergrundbetrieb).
- quantifizieren die Eigenschaften und Auswirkungen auf den periodischen Teil des Echtzeitsystems.
- formulieren die Grundlagen des Slack-Stealing.
- beschreiben den Einsatz von bandweite-bewahrenden Zustellern.
- unterscheiden aufschiebbare Zusteller und Sporadic Server (SpSL und POSIX).
- wenden eine Übernahmeprüfung bei sporadischen Aufgaben mittels dichte- oder schlupfbasierten Akzeptanztests an.
- arbeiten einen strukturierter Ablaufplan (Rahmen) aus und untersuchen den Einsatz von Slack-Stealing.
- ermitteln gerichtete Abhängigkeiten und Rangfolgen in Echtzeitanwendungen (Abhängigkeits- und Aufgabengraph).
- stellen Umsetzungsalternativen für Abhängigkeiten einander gegenüber (naiv, implizit, explizit).
- beschreiben das Konzept der zeitlichen Domänen und physikalischer bzw. logischer Ereignisse.
- übertragen Abhängigkeiten auf das Problem der Ablaufplanung (modifiziere Auslösezeitpunkt/Termin, Phasenversatz).

		• konzipieren Rangfolge und aperiodische Steuerung in eCos. • implementieren einen aperiodischer Moduswechsel mit Zustandsüberführung in eCos. • wenden die Grundlagen von Wettstreit um Betriebsmitteln, Konkurrenz und Konfliktsituationen (kritische Abschnitte, (un)kontrollierte Prioritätsumkehr) an. • beschreiben echtzeitfähige Synchronisationsprotokolle (NPCS, PI, PCP). • nennen die Vor- und Nachteile der Techniken (transitive Blockung, Verklemmungen). • hinterfragen die Vereinfachung des PCP durch stapelbezogene Grenzprioritäten. • bestimmen die Ablaufplanung unter Berücksichtigung von Blockierungszeiten und Selbstsuspendierung. • implementieren Zugriffskontrolle (NPCS, PI, PCP) in Echtzeitanwendungen mit eCos. • erläutern die Anforderungen an verteile Echtzeitsysteme (Komposition, Erweiterbarkeit, Komplexität, Ereignis- vs. Zustandsnachricht). • fassen die Grundlagen von Knoten, Netzwerkschnittstellen und Netzübergängen sowie die Konzepte der expliziten und impliziten Flusskontrolle zusammen. • erschließen sich typische Probleme (zeitliche Analyse, Beobachtbarkeit, Synchronisation, Rangfolge) und Fehlerquellen bei der Programmierung von Echtzeitanwendungen. • können in Gruppen kooperativ und effektiv arbeiten. • können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. • reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab. • können offen und konstruktiv mit Schwachpunkten in der Konzeption wie Umsetzung umgehen.
7	**Voraussetzungen für die Teilnahme**	Für eine erfolgreiche Teilnahme an der Veranstaltung sind grundlegende Programmierkenntnisse in C/C++ erforderlich. Diese können durch den (empfohlenen) Besuch entsprechender Grundlagenveranstaltungen oder im Eigenstudium erworben sein, eine formale Voraussetzung besteht in diesem Zusammenhang nicht.
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	Unregelmäßig

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 60 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Hermann Kopetz. Real-Time Systems: Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers, 1997. • Jane W. S. Liu. Real-Time Systems. Prentice-Hall, Inc., 2000. • Wolfgang Schröder-Preikschat. Softwaresysteme 1. Vorlesungsfolien. 2006.

1	**Modulbezeichnung** 722831	**Middleware - Cloud Computing (Vorlesung mit erweiterten Übungen)** Lecture and advanced tutorial: Middleware - Cloud computing	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Middleware - Cloud Computing (2 SWS) Übung: MW-Ü (2 SWS) Übung: Middleware - Cloud Computing - Übungen (2 SWS) Übung: MW-EÜ (2 SWS)	- - - 5 ECTS
3	Lehrende	PD Dr.Ing. Tobias Distler Harald Böhm Laura Lawniczak	

4	**Modulverantwortliche/r**	PD Dr.Ing. Tobias Distler
5	**Inhalt**	• Überblick Cloud Computing • Grundlagen verteilter Programmierung (Web Services/SOAP/REST) • Virtualisierung als Basis für Cloud Computing • Infrastructure as a Service (IaaS) am Beispiel von Eucalyptus und Amazon EC2 • Skalierbare Verarbeitung von großen Datenmengen • Interoperabilität und Multi-Cloud Computing • Fehlertoleranz im Kontext von Cloud Computing • Aktuelle Forschungstrends
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: - nennen unterschiedliche Ausprägungen von Cloud-Computing. - erläutern verschiedene Cloud-Architekturen. - stellen Vor- und Nachteile von Cloud-Computing gegenüber. - unterscheiden die Herangehensweisen bei der Entwicklung von SOAP- im Vergleich zu REST-Anwendungen. - organisieren den Austausch von Informationen in einer verteilten Anwendung unter Verwendung eines Verzeichnisdienstes. - entwickeln eigene auf Web-Services basierende Anwendungen. - bewerten die Vor- und Nachteile der Bündelung von Nachrichten beim Aufruf von Web-Services. - erläutern die Anforderungen an ein virtualisiertes System. - beschreiben die für die Virtualisierung eines Systems erforderlichen Kriterien. - vergleichen zwischen unterschiedlichen Virtualisierungstechniken und -ebenen. - schildern den Aufbau und die Funktionsweise von Xen und Linux-VServer. - erproben das Einrichten eines Abbilds für eine virtuelle Maschine. - skizzieren die Architektur einer Infrastruktur-Cloud sowie die Aufgabenbereiche hierfür zentraler Komponenten am Beispiel von Eucalyptus.

		- erproben das Bereitstellen von Anwendungen in einer Infrastruktur-Cloud. - zeigen die Grundlagen Software-definierter Netzwerke am Beispiel von Onix und B4 auf. - bewerten verschiedene im Bereich Cloud-Computing zum Einsatz kommende Datenspeichersysteme (Google File System, Bigtable, Windows Azure Storage, Amazon Dynamo) hinsichtlich der Kriterien Verfügbarkeit, Konsistenz und Partitionstoleranz. - erläutern eine auf Vektoruhren basierende Methode zur Auflösung im Zusammenhang mit letztendlicher Konsistenz auftretender Konflikte. - entwickeln ein verteiltes, repliziertes Dateisystem nach dem Vorbild von HDFS, das auf die hierarchische Speicherung großer Datenmengen ausgelegt ist. - erkunden das Bereitstellen selbst entwickelter Dienste mittels Docker. - erstellen ein Framework zur parallelen Bearbeitung von Daten nach dem Vorbild von MapReduce. - konzipieren eigene MapReduce-Anwendungen zur Verarbeitung strukturierter bzw. unstrukturierter Rohdaten. - diskutieren die Fehlertoleranzmechanismen in Google MapReduce. - schildern die grundsätzliche Funktionsweise von Systemen zur Kühlung von Datenzentren mittels Umgebungsluft. - beschreiben das Grundkonzept einer temperaturabhängigen Lastverteilung von Prozessen in einem Datenzentrum. - stellen diverse Ansätze zur Erhöhung der Energieeffizienz von MapReduce-Clustern gegenüber. - unterscheiden die Architekturen und Funktionsweisen der Koordinierungsdienste Chubby und ZooKeeper. - entwickeln einen eigenen Koordinierungsdienst nach dem Vorbild von ZooKeeper. - ermitteln die Konsistenzeigenschaften der eigenen Koordinierungsdienstimplementierung. - erläutern unterschiedliche Ansätze zur Reduzierung bzw. Tolerierung von Tail-Latenz. - skizzieren das Grundkonzept von Erasure-Codes. - beschreiben den Aufbau eines auf die Clouds mehrerer Anbieter gestützten Datenspeichersystems. - erläutern den Einsatz passiver Replikation zur Bereitstellung von Fehlertoleranzmechanismen für virtuelle Maschinen am Beispiel von Remus. - schildern die Grundlagen der Migration von virtuellen Maschinen. - bewerten die Qualität einer aktuellen Publikation aus der Fachliteratur. - erschließen sich typische Probleme (Nebenläufigkeit, Konsistenz, Skalierbarkeit) und Fehlerquellen bei der Programmierung verteilter Anwendungen. - können in Kleingruppen kooperativ arbeiten. - können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. - können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.

		- reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
7	**Voraussetzungen für die Teilnahme**	Gute Programmierkenntnisse in Java
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Datenbanksysteme Bachelor of Science Informatik 20222 Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 557235	**Middleware - Cloud Computing (Vorlesung mit Übungen)** Lecture and tutorial: Middleware - Cloud computing	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Middleware - Cloud Computing (2 SWS) Übung: MW-Ü (2 SWS) Übung: Middleware - Cloud Computing - Übungen (2 SWS)	- - -
3	Lehrende	PD Dr.Ing. Tobias Distler Harald Böhm Laura Lawniczak	

4	**Modulverantwortliche/r**	PD Dr.Ing. Tobias Distler
5	**Inhalt**	• Überblick Cloud Computing • Grundlagen verteilter Programmierung (Web Services/SOAP/REST) • Virtualisierung als Basis für Cloud Computing • Infrastructure as a Service (IaaS) am Beispiel von Eucalyptus und Amazon EC2 • Skalierbare Verarbeitung von großen Datenmengen • Interoperabilität und Multi-Cloud Computing • Fehlertoleranz im Kontext von Cloud Computing • Aktuelle Forschungstrends
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: - nennen unterschiedliche Ausprägungen von Cloud-Computing. - erläutern verschiedene Cloud-Architekturen. - stellen Vor- und Nachteile von Cloud-Computing gegenüber. - unterscheiden die Herangehensweisen bei der Entwicklung von SOAP- im Vergleich zu REST-Anwendungen. - organisieren den Austausch von Informationen in einer verteilten Anwendung unter Verwendung eines Verzeichnisdienstes. - entwickeln eigene auf Web-Services basierende Anwendungen. - erläutern die Anforderungen an ein virtualisiertes System. - beschreiben die für die Virtualisierung eines Systems erforderlichen Kriterien. - vergleichen zwischen unterschiedlichen Virtualisierungstechniken und -ebenen. - schildern den Aufbau und die Funktionsweise von Xen und Linux-VServer. - erproben das Einrichten eines Abbilds für eine virtuelle Maschine. - skizzieren die Architektur einer Infrastruktur-Cloud sowie die Aufgabenbereiche hierfür zentraler Komponenten am Beispiel von Eucalyptus. - erproben das Bereitstellen von Anwendungen in einer Infrastruktur-Cloud. - zeigen die Grundlagen Software-definierter Netzwerke am Beispiel von Onix und B4 auf. - bewerten verschiedene im Bereich Cloud-Computing zum Einsatz kommende Datenspeichersysteme (Google File System, Bigtable,

		Windows Azure Storage, Amazon Dynamo) hinsichtlich der Kriterien Verfügbarkeit, Konsistenz und Partitionstoleranz. - erläutern eine auf Vektoruhren basierende Methode zur Auflösung im Zusammenhang mit letztendlicher Konsistenz auftretender Konflikte. - entwickeln ein verteiltes Dateisystem nach dem Vorbild von HDFS, das auf die Speicherung großer Datenmengen ausgelegt ist. - erkunden das Bereitstellen selbst entwickelter Dienste mittels Docker. - erstellen ein Framework zur parallelen Bearbeitung von Daten nach dem Vorbild von MapReduce. - konzipieren eigene MapReduce-Anwendungen zur Verarbeitung strukturierter Rohdaten. - diskutieren die Fehlertoleranzmechanismen in Google MapReduce. - schildern die grundsätzliche Funktionsweise von Systemen zur Kühlung von Datenzentren mittels Umgebungsluft. - beschreiben das Grundkonzept einer temperaturabhängigen Lastverteilung von Prozessen in einem Datenzentrum. - stellen diverse Ansätze zur Erhöhung der Energieeffizienz von MapReduce-Clustern gegenüber. - unterscheiden die Architekturen und Funktionsweisen der Koordinierungsdienste Chubby und ZooKeeper. - entwickeln einen eigenen Koordinierungsdienst nach dem Vorbild von ZooKeeper. - ermitteln die Konsistenzeigenschaften der eigenen Koordinierungsdienstimplementierung. - erläutern unterschiedliche Ansätze zur Reduzierung bzw. Tolerierung von Tail-Latenz. - skizzieren das Grundkonzept von Erasure-Codes. - beschreiben den Aufbau eines auf die Clouds mehrerer Anbieter gestützten Datenspeichersystems. - erläutern den Einsatz passiver Replikation zur Bereitstellung von Fehlertoleranzmechanismen für virtuelle Maschinen am Beispiel von Remus. - schildern die Grundlagen der Migration von virtuellen Maschinen. - bewerten die Qualität einer aktuellen Publikation aus der Fachliteratur. - erschließen sich typische Probleme (Nebenläufigkeit, Konsistenz, Skalierbarkeit) und Fehlerquellen bei der Programmierung verteilter Anwendungen. - können in Kleingruppen kooperativ arbeiten. - können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. - können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen. - reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.
7	**Voraussetzungen für die Teilnahme**	Gute Programmierkenntnisse in Java
8	**Einpassung in Studienverlaufsplan**	Semester: 5

9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Datenbanksysteme Bachelor of Science Informatik 20222 Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 876012	**Verlässliche Echtzeitsysteme (Vorlesung mit Übungen)** Dependable real-time systems (lecture with exercises)	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Peter Wägemann
5	**Inhalt**	Viele Echtzeitsysteme sind in Bereiche des täglichen Lebens eingebettete, die hohe Anforderungen an die funktionale Sicherheit dieser Systeme stellen. Beispiele hierfür sind Fahrerassistenzsysteme in modernen Automobilen, medizinische Geräte, Prozessanlagen in Kernkraftwerken oder Chemiefabriken oder Flugzeuge. Fehlfunktionen in diesen Anwendungen ziehen mitunter katastrophale Konsequenzen nach sich - Menschen können ernsthaft verletzt oder sogar getötet werden, Landstriche können unbewohnbar gemacht oder zumindest großer finanzieller Schaden verursacht werden. Dieses Modul betrachtet Methoden und Werkzeuge, die uns helfen können, einerseits *zuverlässig Software zu entwickeln* (also Fehler im Programm zu entdecken und zu vermeiden), und andererseits *zuverlässige Software zu entwickeln* (also Abstraktionen, die auch im Fehlerfall ihre Gültigkeit behalten). Hierbei steht weniger die Vermittlung theoretischer Grundkenntnisse auf diesen Gebieten im Vordergrund, also vielmehr • die praktische Anwendung existierende Werkzeuge und Methoden • sowie die Erfahrung und das Verständnis ihrer Grenzen. Auf diese Weise soll ein Fundament für die konstruktive Umsetzung verlässlicher Echtzeitsysteme gelegt werden. Dieses Modul soll daher fundierte Anknüpfungspunkte für die Entwicklung verlässlicher Echtzeitsysteme vermitteln, die Ad-hoc-Techniken möglichst ersetzen sollen.
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: • nennen die Konzepte und die Taxonomie verlässlicher Systeme, unterscheiden Software- und Hardwarefehler und klassifizieren Fehler (Defekt, Fehler, Fehlverhalten). • stellen Fehlerbäume auf. • organisieren Softwareentwicklungsprojekte mittels der Versionsverwaltung git. • vergleichen die verschiedenen Arten der Redundanz als Grundvorraussetzung für Fehlererkennung und -toleranz. • entwickeln fehlertolerante Systeme mittels Replikation. • diskutieren die Fehlerhypothese und die Sicherstellung von Replikdeterminismus. • erläutern die Vor- und Nachteile softwarebasierter Replikation und den Einsatz von Diversität.

		• wenden Informationsredundanz zur Härtung von Daten- und Kontrollflüssen an. • bewerten die Effektivität der arithmetischer Codierung von Programmen und verallgemeinern diesen Ansatz auf die verschiedenen Implementeirungsebenen (Maschinenprogramm zu Prozessinkarnation). • interpretieren den Einfluss der Ausführungsplattform (Hardware, Betriebssystem) auf die Leistungsfähigkeit der Fehlererkennung. • konzipieren eine fehlertolerante Ausführungsumgebung für ein softwarebasiertes TMR-System basierend auf ANBD-Codierung. • nennen die Grundlagen der systematischen Fehlerinjektion. • überprüfen die Wirksamkeit von Fehlertoleranzmechanismen mittels Fehlerinjektion auf der Befehlssatzebene. • entwickeln Testfälle für die Fehlerinjektion mittels des fail** Werkzeugs. • setzten Messergebnisse in Relation zu dem tatsächlichen Fehlerraum. • beschreiben die Grundlagen der Fehlererholung (Vorwärts- bzw. Rückwärtskorrektur) und Reintegration fehlgeschlagener Knoten. • vergleichen den Zustandstransfer am Beispiel der Running bzw. Recursive State Restoration. • benennen Konzepte der Rückwärtskorrektur durch Entwurfsalternativen (Recovery Blocks). • fassen die Grundlagen des dynamischen Testens zusammen. • unterscheiden Black-Box und White-Box Testverfahren. • konzipieren und implementieren Testfälle. • überprüfen die Testüberdeckung anhand grundlegender Überdeckungskriterien (Anweisungs- bis Bedingungsüberdeckung). • geben die Grundlagen der statischen Programmanalyse wieder. • nennen die Funktionsweise von Hoare- WP-Kalkül. • verifizieren eine Ampelsteuerung mittels des FramaC Werkzeugs zur statischen Analyse von C Programmen. • beschreiben den Korrektheitsnachweis mittels abstrakter Interpretation und unterscheiden die konkrete von der abstrakten Programmsemnatik. • erläutern die Funktionsweise von Sammel- und Präfixsemantiken. • erstellen einen Korrektheitsbeweis für einen a-b-Filter mittels des Astrée Werkzeugs zur abstrakten Interpretation von C Programmen. • bewerten die Verlässlichkeit kommerzieller, sicherheitskritischer Systeme anhand von Fallstudien (Sizewell B, Airbus A320).

		• erschließen sich typische Probleme und Fehlerquellen bei der Programmierung von eingebetteten Systemen im Allgemeinen. • klassifizieren Fallstricke und Mehrdeutigkeiten in der Programmiersprache C99 im Besonderen. • können in Gruppen kooperativ und effektiv arbeiten. • können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. • reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab. • können offen und konstruktiv mit Schwachpunkten in der Konzeption wie Umsetzung umgehen.
7	**Voraussetzungen für die Teilnahme**	Für eine erfolgreiche Teilnahme an der Veranstaltung sind grundlegende Programmierkenntnisse in C/C++ erforderlich. Diese können durch den (empfohlenen) Besuch entsprechender Grundlagenveranstaltungen oder im Eigenstudium erworben sein. Weiterhin sind grundlegende Kenntnisse über Echtzeitsystemeeine, zum Beispiel durch den Besuch der Veranstaltung "Echtzeitsysteme", empfohlen. Eine formale Voraussetzung besteht in diesem Zusammenhang jedoch nicht.
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 743260	**Verteilte Systeme (Vorlesung mit erweiterten Übungen)** Distributed systems (lecture with extended exercises)	**7,5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	PD Dr.Ing. Tobias Distler
5	**Inhalt**	Verteilte Systeme bestehen aus mehreren Rechnern, die über ein Netzwerk miteinander verbunden sind und einen gemeinsamen Dienst erbringen. Obwohl die beteiligten Rechner hierfür in weiten Teilen unabhängig voneinander agieren, erscheinen sie ihren Nutzern gegenüber in der Gesamtheit dabei trotzdem als ein einheitliches System. Die Einsatzmöglichkeiten für verteilte Systeme erstrecken sich über ein weites Spektrum an Szenarien: Von der Zusammenschaltung kleinster Rechenknoten zur Sammlung von Daten im Rahmen von Sensornetzwerken über Steuerungssysteme für Kraftfahrzeuge und Industrieanlagen bis hin zu weltumspannenden, Internet-gestützten Infrastrukturen mit Komponenten in Datenzentren auf verschiedenen Kontinenten. Ziel dieses Moduls ist es, die sich durch die speziellen Eigenschaften verteilter Systeme ergebenden Problemstellungen zu verdeutlichen und Ansätze zu vermitteln, mit deren Hilfe sie gelöst werden können; Beispiele hierfür sind etwa die Interaktion zwischen heterogenen Systemkomponenten, der Umgang mit erhöhten Netzwerklatenzen sowie die Wahrung konsistenter Zustände über Rechnergrenzen hinweg. Gleichzeitig zeigt das Modul auf, dass die Verteiltheit eines Systems nicht nur Herausforderungen mit sich bringt, sondern auf der anderen Seite auch Chancen eröffnet. Dies gilt insbesondere in Bezug auf die im Vergleich zu nicht verteilten Systemen erzielbare höhere Widerstandsfähigkeit eines Gesamtsystems gegenüber Fehlern wie den Ausfällen ganzer Rechner oder sogar kompletter Datenzentren. Ausgehend von den einfachsten, aus nur einem Client und einem Server bestehenden verteilten Systemen, beschäftigt sich die Vorlesung danach mit der deutlich komplexeren Replikation der Server-Seite und behandelt anschließend die Verteilung eines Systems über mehrere, mitunter weit voneinander entfernte geografische Standorte. In allen Abschnitten umfasst die Betrachtung des jeweiligen Themas eine Auswahl aus Grundlagen, im Praxiseinsatz befindlicher Ansätze und Techniken sowie für den aktuellen Stand der Forschung repräsentativer Konzepte. Im Rahmen der Übungen wird zunächst ein plattformunabhängiges Fernaufrufsystem schrittweise entwickelt und parallel dazu getestet. Als Vorlage und Orientierungshilfe dient dabei das in der Praxis weit verbreitete Java RMI. In den weiteren Übungsaufgaben stehen anschließend klassische Problemstellungen von verteilten Systemen wie fehlertolerante Replikation und verteilte Synchronisation im Mittelpunkt.

6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: • beschreiben charakteristische Merkmale und Eigenschaften verteilter Systeme sowie grundlegende Probleme im Zusammenhang mit ihrer Realisierung. • untersuchen die Unterschiede zwischen lokalen Methodenaufrufen und Fernmethodenaufrufen. • vergleichen Ansätze zur Konvertierung von Nachrichten zwischen verschiedenen Datenrepräsenationen. • konzipieren eine eigene auf Java RMI basierende Anwendung. • entwickeln ein eigenes Fernaufrufsystem nach dem Vorbild von Java RMI. • bewerten die Serialisierungsroutinen von Java RMI. • erproben die manuelle Serialisierung von Nachrichten. • bewerten die Performanz des eigenen Fernaufrufsystems. • gestalten ein Modul zur Unterstützung verschiedener Fernaufrufsemantiken (Maybe, Last-of-Many, At-Most-Once) für das eigene Fernaufrufsystem. • beurteilen auf Basis eigener Experimente mit Fehlerinjektionen die Auswirkungen von Störeinflüssen auf verschiedene Fernaufrufsemantiken. • klassifizieren Mechanismen zur Bereitstellung von Fehlertoleranz, insbesondere verschiedene Arten der Replikation (aktiv vs. passiv). • vergleichen verschiedene Konsistenzgarantien georeplizierter Systeme. • illustrieren das Problem einer fehlenden gemeinsamen Zeitbasis in verteilten Systemen. • erforschen logische Uhren als Mittel zur Reihenfolgebestimmung und Methoden zur Synchronisation physikalischer Uhren. • unterscheiden grundlegende Zustellungs- und Ordnungsgarantien beim Multicast von Nachrichten. • gestalten ein Protokoll für den zuverlässigen und totalgeordneten Versand von Nachrichten in einer Gruppe von Knoten. • entwickeln einen Dienst zur Verwaltung verteilter Sperrobjekte auf Basis von Lamport-Locks. • bewerten die Qualität einer Publikation aus der Fachliteratur. • erschließen sich typische Probleme (Nebenläufigkeit, Konsistenz) und Fehlerquellen bei der Programmierung verteilter Anwendungen. • können in Kleingruppen kooperativ arbeiten. • können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. • reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab. • können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen.

7	**Voraussetzungen für die Teilnahme**	Gute Programmierkenntnisse in Java
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 649073	**Verteilte Systeme (Vorlesung mit Übungen)** Lecture and tutorial: Distributed systems	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	PD Dr.Ing. Tobias Distler
5	**Inhalt**	Verteilte Systeme bestehen aus mehreren Rechnern, die über ein Netzwerk miteinander verbunden sind und einen gemeinsamen Dienst erbringen. Obwohl die beteiligten Rechner hierfür in weiten Teilen unabhängig voneinander agieren, erscheinen sie ihren Nutzern gegenüber in der Gesamtheit dabei trotzdem als ein einheitliches System. Die Einsatzmöglichkeiten für verteilte Systeme erstrecken sich über ein weites Spektrum an Szenarien: Von der Zusammenschaltung kleinster Rechenknoten zur Sammlung von Daten im Rahmen von Sensornetzwerken über Steuerungssysteme für Kraftfahrzeuge und Industrieanlagen bis hin zu weltumspannenden, Internet-gestützten Infrastrukturen mit Komponenten in Datenzentren auf verschiedenen Kontinenten. Ziel dieses Moduls ist es, die sich durch die speziellen Eigenschaften verteilter Systeme ergebenden Problemstellungen zu verdeutlichen und Ansätze zu vermitteln, mit deren Hilfe sie gelöst werden können; Beispiele hierfür sind etwa die Interaktion zwischen heterogenen Systemkomponenten, der Umgang mit erhöhten Netzwerklatenzen sowie die Wahrung konsistenter Zustände über Rechnergrenzen hinweg. Gleichzeitig zeigt das Modul auf, dass die Verteiltheit eines Systems nicht nur Herausforderungen mit sich bringt, sondern auf der anderen Seite auch Chancen eröffnet. Dies gilt insbesondere in Bezug auf die im Vergleich zu nicht verteilten Systemen erzielbare höhere Widerstandsfähigkeit eines Gesamtsystems gegenüber Fehlern wie den Ausfällen ganzer Rechner oder sogar kompletter Datenzentren. Ausgehend von den einfachsten, aus nur einem Client und einem Server bestehenden verteilten Systemen, beschäftigt sich die Vorlesung danach mit der deutlich komplexeren Replikation der Server-Seite und behandelt anschließend die Verteilung eines Systems über mehrere, mitunter weit voneinander entfernte geografische Standorte. In allen Abschnitten umfasst die Betrachtung des jeweiligen Themas eine Auswahl aus Grundlagen, im Praxiseinsatz befindlicher Ansätze und Techniken sowie für den aktuellen Stand der Forschung repräsentativer Konzepte. Im Rahmen der Übungen wird zunächst ein plattformunabhängiges Fernaufrufsystem schrittweise entwickelt und parallel dazu getestet. Als Vorlage und Orientierungshilfe dient dabei das in der Praxis weit verbreitete Java RMI. In den weiteren Übungsaufgaben stehen anschließend klassische Problemstellungen von verteilten Systemen

		wie fehlertolerante Replikation und verteilte Synchronisation im Mittelpunkt.
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: • beschreiben charakteristische Merkmale und Eigenschaften verteilter Systeme sowie grundlegende Probleme im Zusammenhang mit ihrer Realisierung. • untersuchen die Unterschiede zwischen lokalen Methodenaufrufen und Fernmethodenaufrufen. • vergleichen Ansätze zur Konvertierung von Nachrichten zwischen verschiedenen Datenrepräsenationen. • konzipieren eine eigene auf Java RMI basierende Anwendung. • entwickeln ein eigenes Fernaufrufsystem nach dem Vorbild von Java RMI. • gestalten ein Modul zur Unterstützung verschiedener Fernaufrufsemantiken (Maybe, Last-of-Many) für das eigene Fernaufrufsystem. • beurteilen auf Basis eigener Experimente mit Fehlerinjektionen die Auswirkungen von Störeinflüssen auf verschiedene Fernaufrufsemantiken. • klassifizieren Mechanismen zur Bereitstellung von Fehlertoleranz, insbesondere verschiedene Arten der Replikation (aktiv vs. passiv). • vergleichen verschiedene Konsistenzgarantien georeplizierter Systeme. • illustrieren das Problem einer fehlenden gemeinsamen Zeitbasis in verteilten Systemen. • erforschen logische Uhren als Mittel zur Reihenfolgebestimmung und Methoden zur Synchronisation physikalischer Uhren. • unterscheiden grundlegende Zustellungs- und Ordnungsgarantien beim Multicast von Nachrichten. • gestalten ein Protokoll für den zuverlässigen und totalgeordneten Versand von Nachrichten in einer Gruppe von Knoten. • entwickeln einen Dienst zur Verwaltung verteilter Sperrobjekte auf Basis von Lamport-Locks. • bewerten die Qualität einer Publikation aus der Fachliteratur. • erschließen sich typische Probleme (Nebenläufigkeit, Konsistenz) und Fehlerquellen bei der Programmierung verteilter Anwendungen. • können in Kleingruppen kooperativ arbeiten. • können ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten. • können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen. • reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab.

7	**Voraussetzungen für die Teilnahme**	Gute Programmierkenntnisse in Java
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 202041	**Virtuelle Maschinen (Vorlesung mit Übungen)** Lecture and tutorial: Virtual machines	**5 ECTS**
2	Lehrveranstaltungen	Übung: Virtuelle Maschinen - Übung (2 SWS) Vorlesung: Virtuelle Maschinen (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Dr.-Ing. Volkmar Sieh	

4	**Modulverantwortliche/r**	Dr.-Ing. Volkmar Sieh
5	**Inhalt**	Vorgestellt werden verschiedene Virtualisierungs-Ansätze: • Emulation • Just-In-Time-Compiler • Para-Virtualisierung • Bibliotheks-basierte Virtualisierung • OS-Virtualisierung
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: • erläutern verschiedene Motivationen für den Einsatz von VMs • unterscheiden verschiedene VMs • klassifizieren verschiedene Ziele unterschiedlicher VMs (z.B. Performance, Konfigurierbarkeit, Genauigkeit, ...) • hinterfragen verschiedene Simulationansätze für MMUs • erstellen virtuelle Komponenten und Busse • strukturieren Callbacks und entsprechendes Forwarding und Caching • unterscheiden zwischen Architektur, Chip und Komponente • klassifizieren unterschiedliche Just-In-Time-Compiler-Ansätze • erzeugen JIT Code aus vorgefertigten Code-Teilen • bewerten unterschiedliche JIT-Code-Optimierungen • erläutern Probleme bei der JIT-Code-Invalidierung • nennen JIT Probleme mit Exceptions/Interrupts sowie berechnete Sprüngen und Return-Instruktionen • unterscheiden verschiedene JIT Cache-Verwaltungen • beschreiben Möglichkeiten der Fehlerinjektion durch VMs • entwickeln ein an JIT angepasstes virtuelles "Hardware"-Design • erläutern die Java-VM Instruktionssatz-Architektur • nutzen Hardware-basierte Virtualisierung • entwickeln Verfahren zum Ausfiltern bestimmter Befehle • erläutern Probleme der Speicherverwaltung bei HW-basierter Virtualisierung • nutzen User-Mode-Emulation zur Paravirtualisierung • diskutieren Möglichkeiten von Debuggern für die Umleitung von System-Calls und die Ausfilterung von Befehlen • nutzen einen Hypervisor zur Paravirtualisierung • unterscheiden verschiedene Ansätze zur Geräteverwaltung in paravirtualisierten Systemen • erläutern Betriebssystem-basierte Virtualisierung • entwickeln unterschiedliche Bibliotheks-basierte Virtualisierungen

		• erläutern Probleme beim Speicher-Layout bei Bibliotheks-basierte Virtualisierung • konzipieren Personalities für Bibliotheks-basierte Virtualisierungen • beurteilen Probleme bei der korrekten Zeit-Simulation • nennen Ideen für die dynamische Anpassung der Zeit-Simulation • klassifizieren bekannte VMs (z.B. VICE, FAUmachine, QEMU, Bochs, JVM, KVM, User-Mode-Linux, Xen, VServer, Wine) • diskutieren in der Gruppe Vor- und Nachteile von bestimmten VM-Ansätzen • untersuchen CPU-Emulationen • untersuchen Geräte-Emulationen
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Rechnerarchitektur Bachelor of Science Informatik 20222 Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 462793	**Virtuelle Maschinen (Vorlesung mit Übung und Laborübung)** Lecture, tutorial and laboratory: Virtual machines	**7,5 ECTS**
2	Lehrveranstaltungen	Übung: VM-EÜ (2 SWS) Übung: Virtuelle Maschinen - Übung (2 SWS) Vorlesung: Virtuelle Maschinen (2 SWS)	2,5 ECTS 2,5 ECTS 2,5 ECTS
3	Lehrende	Dr.-Ing. Volkmar Sieh	

4	**Modulverantwortliche/r**	Dr.-Ing. Volkmar Sieh
5	**Inhalt**	Vorgestellt werden verschiedene Virtualisierungs-Ansätze: • Emulation • Just-In-Time-Compiler • Para-Virtualisierung • Bibliotheks-basierte Virtualisierung • OS-Virtualisierung
6	**Lernziele und Kompetenzen**	Studierende, die das Modul erfolgreich abgeschlossen haben: • erläutern verschiedene Motivationen für den Einsatz von VMs • unterscheiden verschiedene VMs • klassifizieren verschiedene Ziele unterschiedlicher VMs (z.B. Performance, Konfigurierbarkeit, Genauigkeit, ...) • hinterfragen verschiedene Simulationansätze für MMUs • erstellen virtuelle Komponenten und Busse • strukturieren Callbacks und entsprechendes Forwarding und Caching • unterscheiden zwischen Architektur, Chip und Komponente • klassifizieren unterschiedliche Just-In-Time-Compiler-Ansätze • erzeugen JIT Code aus vorgefertigten Code-Teilen • bewerten unterschiedliche JIT-Code-Optimierungen • erläutern Probleme bei der JIT-Code-Invalidierung • nennen JIT Probleme mit Exceptions/Interrupts sowie berechnete Sprüngen und Return-Instruktionen • unterscheiden verschiedene JIT Cache-Verwaltungen • beschreiben Möglichkeiten der Fehlerinjektion durch VMs • entwickeln ein an JIT angepasstes virtuelles "Hardware"-Design • erläutern die Java-VM Instruktionssatz-Architektur • nutzen Hardware-basierte Virtualisierung • entwickeln Verfahren zum Ausfiltern bestimmter Befehle • erläutern Probleme der Speicherverwaltung bei HW-basierter Virtualisierung • nutzen User-Mode-Emulation zur Paravirtualisierung • diskutieren Möglichkeiten von Debuggern für die Umleitung von System-Calls und die Ausfilterung von Befehlen • nutzen einen Hypervisor zur Paravirtualisierung • unterscheiden verschiedene Ansätze zur Geräteverwaltung in paravirtualisierten Systemen • erläutern Betriebssystem-basierte Virtualisierung

		• entwickeln unterschiedliche Bibliotheks-basierte Virtualisierungen • erläutern Probleme beim Speicher-Layout bei Bibliotheks-basierte Virtualisierung • konzipieren Personalities für Bibliotheks-basierte Virtualisierungen • beurteilen Probleme bei der korrekten Zeit-Simulation • nennen Ideen für die dynamische Anpassung der Zeit-Simulation • klassifizieren bekannte VMs (z.B. VICE, FAUmachine, QEMU, Bochs, JVM, KVM, User-Mode-Linux, Xen, VServer, Wine) • diskutieren in der Gruppe Vor- und Nachteile von bestimmten VM-Ansätzen • entwickeln selbst CPU-Emulationen • entwickeln selbst Geräte-Emulationen • verteilen Implementierungsaufgaben in ihrer Gruppe • erstellen Zeitpläne für Implementierungen für sich und ihre Gruppe
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Rechnerarchitektur Bachelor of Science Informatik 20222 Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93087	**Web-basierte Systeme** Web-based systems	**5 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Rüdiger Kapitza
5	**Inhalt**	Die Vorlesung vermittelt grundsätzliches Verständnis von Web-basierten Systemen. Dies schließt Basistechnologien, wie das HTTP-Protokoll ein, sowie XML und HTML als Mittel zur Informationsbeschreibung und -darstellung. Weiterhin werden verschiedene Dienstarchitekturen vorgestellt und sowohl Server- als auch Client-seitige Programmierung von Web-basierten Systemen erarbeitet. Studierende lernen somit den Entwurf und die Implementierung von Web-basierten Anwendungen. • Technische Grundlagen des Internets • HTTP als Transportprotokoll • XML und HTML • Paradigmenwechsel zu Web-basierten Systemen • Architektur Web-basierter Systeme • Serverseitige Implementierung von Web-basierten Systemen • Skalierbare Serverdienste • Clientseitige Programmierung von aktiven Inhalten (Bspw. mit JavaScript) • Architektur moderner Browser • Peer-to-Peer basierte Browseranwendungen
6	**Lernziele und Kompetenzen**	Grundsätzliches Verständnis von Web-basierten Systemen. Dies schließt Basistechnologien, wie das HTTP-Protokoll ein, sowie XML und HTML als Mittel zur Informationsbeschreibung und -darstellung. Weiterhin werden verschiedene Dienstarchitekturen vorgestellt und sowohl Server- als auch Client-seitige Programmierung von Web-basierten Systemen erarbeitet. Studierende lernen somit den Entwurf und die Implementierung von Web-basierten Anwendungen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 2
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Datenbanksysteme Bachelor of Science Informatik 20222 Vertiefungsrichtung Verteilte Systeme und Betriebssysteme Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)

12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	
16	**Literaturhinweise**	High Performance Browser Networking What every web developer should know about networking and web performance, O'Reilly Media, 2013 Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Modern JS Libraries, O'Reilly Media, 2014 Weitere Literatur wird auf der Webseite zur Veranstaltung angeboten

Vertiefungsrichtung IT-Sicherheit

1	**Modulbezeichnung** 326311	**Angewandte IT-Sicherheit** Applied IT security	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Angewandte IT-Sicherheit (2 SWS)	2,5 ECTS
3	Lehrende	Dr.Ing. Ralph Palutke	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Felix Freiling
5	**Inhalt**	Die Vorlesung gibt einen einführenden Überblick über Konzepte und Methoden der IT-Sicherheit und eignet sich als Einstieg in das Vertiefungsgebiet "IT-Sicherheit" an der FAU. Themen (unter anderem): IT-Sicherheit vs. physische Sicherheit, Identifizierung und Authentifizierung, grundlegende Muster von Unsicherheit in technischen Systemen, grundlegende Abwehrmechanismen, ausgewählte Beispiele aus dem Bereich Systemsicherheit, Netzwerksicherheit und Softwaresicherheit. In der Übung werden die Themen der Veranstaltung beispielhaft eingeübt. Themen (unter anderem): "lock picking", "social engineering", ausnutzen von Softwareschwachstellen. **Wichtiger Hinweis:** Ab dem Wintersemester 2022/23 wird die **neue Pflichtvorlesung "Sichere Systeme"** (1. Semester, Bachelor Informatik) die Einstiegsvorlesung in den Bereich IT-Sicherheit an der FAU sein. In dieser Rolle ersetzt sie sowohl "Angewandte IT-Sicherheit" (AppITSec) als auch "Einführung in die IT-Sicherheit" (EinfITSec). **Für Informatik-Studierende, die AppITSec noch belegen wollen**, wird auf absehbare Zeit (mindestens bis Sommersemester 2024) die Prüfung in AppITSec angeboten. Die Vorlesungs- und Übungsinhalte sind über einen offenen StudON-Kurs zugreifbar. Bitte nutzen Sie für Fragen das dortige Forum oder die Sprechstunden der Veranstalter. Im Wintersemester 2022/2023 wird eine schriftliche Klausur über 90 Minuten angeboten. Link zum StudON-Kurs: *https://www.studon.fau.de/crs4774802.html*
6	**Lernziele und Kompetenzen**	Teilnehmer erwerben einen Überblick über Konzepte und Methoden aus dem Bereich der IT-Sicherheit und können diese im Kontext der Informatik und der Lebenswirklichkeit anhand von Beispielen einordnen und erläutern. Die Studierenden können die wichtigsten Arten von Softwareschwachstellen in Programmen erkennen und benennen. Sie können außerdem erläutern, wie man diese Schwachstellen ausnutzt und welche technischen und organisatorischen Maßnahmen geeignet sind, diese Schwachstellen zu vermeiden. Die Studierenden lernen, die Wirksamkeit von IT-Sicherheitsmechanismen im gesellschaftlichen Kontext und in Kenntnis professioneller Strukturen der Cyberkriminalität aus technischen, ethischen und rechtlichen Perspektiven zu bewerten.
7	**Voraussetzungen für die Teilnahme**	Keine

8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung IT-Sicherheit Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel (90 Minuten)
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Dieter Gollmann: Computer Security. 3. Auflage, Wiley, 2010. Joachim Biskup: Security in Computing Systems. Springer, 2008. Weitere Literatur wird in der Vorlesung bekanntgegeben.

1	**Modulbezeichnung** 792501	**Forensische Informatik** Forensic computing (lecture with tutorial)	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Felix Freiling
5	**Inhalt**	Forensische Informatik befasst sich mit der Sammlung, Aufbereitung und Analyse digitaler Beweismittel zur Verwendung vor Gericht. Ausgangspunkt ist jeweils der Verdacht auf einen Computereinbruch oder eine Straftat, die mit Hilfe von digitalen Geräten vorgenommen worden ist. Dieses Modul gibt einen Überblick über die Methoden der forensischen Informatik aus einer wissenschaftlichen Perspektive. Der Schwerpunkt liegt auf der Analyse von Dateisystemen. Ziel der Lehrveranstaltung ist nicht die Ausbildung von Forensik-Praktikern, sondern die Vermittlung von Kenntnissen, die es einem erlauben, Forschung im Bereich Computerforensik zu betreiben. Im Rahmen der Übung werden die Themen der Vorlesung im Rahmen von Fallstudien praktisch eingeübt. Voraussichtliche Themen: • Definition forensische Informatik • Der forensische Prozess und seine wissenschaftliche Fundierung • Rechtliche Rahmenbedingungen • Sichern von Festplatten • Analyse verschiedener Dateisysteme (FAT32, NTFS, Ext2/ Ext3) • Tools
6	**Lernziele und Kompetenzen**	Die Studierenden können Termini und Methoden der digitalen Forensik in die Entwicklung der forensischen Wissenschaften einordnen. Die Studierenden können die wesentlichen Datenstrukturen verschiedener Dateisysteme erklären. Sie können die für forensische Zwecke wesentlichen Datenstrukturen lokalisieren und geeignete Werkzeuge zu ihrer Analyse auswählen und anwenden. Die Studierenden können digitale Spuren konkreter Fallkonstellationen durch Anwendung von Werkzeugen rekonstruieren, analysieren, interpretieren und dokumentieren. Sie lernen ihre Untersuchungsergebnisse zu präsentieren und gegenüber kritischen Nachfragen zu verteidigen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung IT-Sicherheit Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	mündlich (30 Minuten) Die mündliche Prüfung dauert 30 Minuten.
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Brian Carrier: File System Forensic Analysis. Addison-Wesley, 2005. • Eoghan Casey: Digital Evidence and Computer Crime - Forensic Science, Computers and the Internet, 3rd Edition. Academic Press 2011. • Andreas Dewald, Felix Freiling: Forensische Informatik. 3. Auflage, BoD, 2015.

1	**Modulbezeichnung** 658644	**Human Factors in Security and Privacy** Human factors in security and privacy	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	PD Dr.habil. Zinaida Benenson
5	**Inhalt**	This course provides insight into the ways in which people interact with IT security. Special attention will be paid to complex environments such as companies, governmental organizations or hospitals. A number of guest talks from practitioners and researchers highlight some of the issues in greater depth. The course covers the following topics: • Terminology of security and privacy, technical and non-technical protection measures • Development and testing of usable security mechanisms (encryption and authentication tools, security policies, security warnings) • Risk perception and decision making in security and privacy context (usage of security software, reaction to security warnings, divulging information in social media) • Economics approach to security and privacy decision making (traditional and behavioral economics) • Trade-offs between the national security and surveillance (psychology behind the EU data retention directive and NSA programs) • Psychological principles of cyber fraud (scams, phishing, social engineering) • Security awareness and user education • Interplay of safety and security in complex systems • Research methods in human factors (qualitative vs. quantitative research, usability testing, experimental design, survey design, interviews) The exercises aim at deepening the understanding of the topics and are highly relevant for examinations. We plan to conduct approximately 5-6 exercises per semester; the rest of the exercises is reserved for the guest talks. A typical exercise consist of two parts: (1) For each topic, the students receive a homework assignment consisting of practical exercises. (2) For each topic, the students receive 1-3 papers to read for the next exercise. The papers will be discussed in the class with the teaching assistant.
6	**Lernziele und Kompetenzen**	Students develop a mindset that naturally takes into account typical psychological and physical characteristics of the users when developing or evaluating security- and privacy-enhancing technologies or policies. Students can: • define terms "security and "privacy

		• identify main research questions in the area of human factors in security and privacy • demonstrate specific difficulties in developing and testing of usable security mechanisms • explain main psychological principles behind the cyber fraud • illustrate specific difficulties in awareness campaigns and user training in the realms of security and privacy • illustrate the influence of the psychological risk perception principles (especially under- and overestimation of risk) on security and privacy decision making • compare different approaches to the development of usable security features • apply elements of the mental models approach and of user-centered design to development and evaluation of security- and privacy-enhancing techniques • scan research papers and other materials for important points that clarify and deepen course contents • structure the relation between usability and security • contrast the approaches of traditional and behavioral economics to the explanation of security- and privacy-related behavior • argue advantages and disadvantages of mass surveillance and other kinds of mass data collection for security and privacy of citizens • critically appraise design and results of published user studies • critically appraise technological solutions or policies for likely "human factors weaknesses in design and usage • develop well-founded personal opinions on the course topics and defend them in the class discussions
7	**Voraussetzungen für die Teilnahme**	LANGUAGE: This module will be held in German. Slides and all other written materials are in English. Assignments and exams are in English and can be answered in English or German. REQUIRED SKILLS: basic knowledge in the area of IT security and privacy, such as security goals (CIA), basic protection mechanisms (symmetric and asymmetric cryptography principles), cryptographic hash functions, digital certificates, PKI, basics of SSL/TLS. This knowledge can be acquired through the attendance of the module "Applied IT Security or similar modules.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung IT-Sicherheit Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester

13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	We use classical and current research papers on usable security and privacy that will be introduced during the module.

1	**Modulbezeichnung** 327615	**Security and Privacy in Pervasive Computing** Security and privacy in pervasive computing (lecture with exercises)	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: SecPriPC (2 SWS) Übung: Security and Privacy in Pervasive Computing - Übung (2 SWS)	- -
3	Lehrende	PD Dr.habil. Zinaida Benenson	

4	**Modulverantwortliche/r**	PD Dr.habil. Zinaida Benenson
5	**Inhalt**	Pervasive Computing, also called Ubiquitous Computing, is a computing paradigm that comprises billions computing devices integrated into everyday objects and connected into a global communication network that is orders of magnitude larger than the Internet today. These devices measure environmental characteristics, exchange information about their surroundings and interact with people in many different ways, such that sometimes people may be even unaware that they are using computers. The era of pervasive computing has already started and moves on rapidly, integrating the Internet, smartphones, wearable computing devices (such as Google glass or Apple Watch), smart grid, home automation, intelligent cars and smart cities. In this course we look at the visions and current scenarios of Pervasive Computing from the security and privacy point of view. We consider security mechanisms and privacy concerns of the present-day technologies, such as smartphone operating systems, GSM/UMTS, WLAN, Bluetooth, ZigBee, RFID, and also of present and envisioned systems and services such vehicular networks, sensor networks, location-based services and augmented reality. The exercise comprises (1) practical tasks on specific attacks, such as eavesdropping on WiFi or ZigBee communication, and (2) guest talks on selected topics, for example, NFC security. For practical exercises, students will be divided into groups, and each group will have to execute the tasks in our lab and write a report about their work for each task. Further details will be communicated in the first exercise.
6	**Lernziele und Kompetenzen**	The students are able to: • recognize existing and future computing systems as pervasive through analysis of their conceptual design and development, deployment and actual usage • critically appraise pervasive computing systems for typical security- and privacy-related concerns and weaknesses in design, deployment and usage • choose appropriate techniques and policies for securing pervasive computing systems • choose appropriate techniques and policies for addressing privacy issues in pervasive computing systems

7	**Voraussetzungen für die Teilnahme**	LANGUAGE: This module will be held in German. Slides and all other written materials are in English. Assignments and exams are in English and can be answered in English or German. REQUIRED SKILLS: Basic knowledge in the area of IT security and privacy, for example: security goals (CIA), symmetric and asymmetric cryptography principles, PKI, basics of SSL/TLS and other security protocols. This knowledge can be acquired through the attendance of the module "Applied IT Security (Angewandte IT Sicherheit) or similar modules.
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung IT-Sicherheit Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Books and papers will be presented during the lecture.

1	**Modulbezeichnung** 172338	**Security in Embedded Hardware** Security in embedded hardware	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Joachim Falk Prof. Dr.-Ing. Jürgen Teich
5	**Inhalt**	Der Schutz eingebetteter Systeme gegenüber Angriffe Dritter auf gespeicherte Daten und Implementierungen, stellt eine immer wichtigere, jedoch auch durch zunehmende Vernetzung herausfordernde Aufgabe dar. Der Schutz der eingebetteten Systeme gegenüber bekannten als auch neueren ausgeklügelten Angriffsmöglichkeiten ist Gegenstand dieser Vorlesung. Es wird gezeigt, welche Angriffe existieren, welche Gegenmaßnahmen man ergreifen kann und wie man sichere eingebettete Systeme entwirft. Einleitung und Motivation • Was ist Security? • Die Bedeutung von Security für zuverlässige Systeme • Klassifikation von Angriffen • Entwurf eingebetteter Systeme Angriffsszenarien • Beispiele von Angriffsszenarien • Kryptographischer Algorithmen als Ziel von Angriffen Angriffe durch Einschleusen von Code (Code Injection Attacks) • Welche Arten von Code Injection-Angriffe gibt es? • Gegenmaßnahmen Invasive physikalische Angriffe (Invasive Physical Attacks) • Microprobing • Reverse Engineering • Differential Fault Analysis • Gegenmaßnahmen Nichtinvasive softwarebasierte Angriffe (Non-Invasive Logical Attacks) • Erlangen von nicht autorisiertem Zugriff • Gegenmaßnahmen Nichtinvasive physikalische Angriffe (Non-Invasive Physical Attacks) • Abhören • Seitenkanalangriffe • Gegenmaßnahmen
6	**Lernziele und Kompetenzen**	Fachkompetenz - Wissen • Die Studierenden legen die entsprechenden Gegenmaßnahmen dar • Die Studierenden nennen verschiedene Sicherheitseinrichtungen und -maßnahmen in eingebetteten Systemen Fachkompetenz - Verstehen

		• Die Studierenden zeigen den Einfluss von Angriffen und deren Gegenmaßnahmen auf die Verlässlichkeit eines eingebetteten Systems auf • Die Studierenden zeigen den zusätzlichen Aufwand (Fläche, Rechenzeit) von Sicherheitseinrichtungen auf Fachkompetenz - Analysieren • Die Studierenden klassifizieren verschiedene Angriffstypen auf eingebettete Systeme Sozialkompetenz • Die Studierenden erarbeiten kooperativ in Gruppen Lösungskonzepte und implementieren diese gemeinsam
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Hardware-Software-Co-Design Bachelor of Science Informatik 20222 Vertiefungsrichtung IT-Sicherheit Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Empfohlene Bücher zur Begleitung und Vertiefung: • Catherine H. Gebotys Security in Embedded Devices. Springer 2010. • Benoit Badrignans et al. Security Trends for FPGAs. Springer 2011. • Daniel Ziener Techniques for Increasing Security and Reliability of IP Cores Embedded in FPGA and ASIC Designs. Dr. Hut 2010. Weitere Informationen:

		https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/security-in-embedded-hardware

1	**Modulbezeichnung** 93098	**Software Exploitation**	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Felix Freiling
5	**Inhalt**	Software verfügt aufgrund ihrer Komplexität häufig unbekannte bzw. unerwünschte Zusatzfunktionalität, die durch geschickte Eingaben provoziert werden kann. Die Ursachen solcher Funktionalität werden als (Software-)Schwachstellen bezeichnet. Beschreibungen von Eingaben, die diese Funktionalität auslösen, nennt man Exploits. Software Exploitation umfasst demnach die Suche nach Schwachstellen in Software und die Erstellung von Exploits für diese Schwachstellen. Die Vorlesung gibt einen Überblick über verbreitete Klassen von Schwachstellen in Software und wie man sie ausnutzen kann. Der Einsatz der vermittelten Kenntnisse und Kompetenzen im Kontext von Ethik und Recht wird ebenfalls angesprochen. Begleitet wird die Vorlesung von Übungen, in denen die vorgestellten Konzepte von den Studierenden praktisch umgesetzt und vertieft werden. Dazu werden Übungsaufgaben gestellt, die nach einer Bearbeitungszeit von jeweils einer Woche gemeinsam mit den Übungsgruppenleitern besprochen werden.
6	**Lernziele und Kompetenzen**	• typische Schwachstellen in Quell- und Binärcode erkennen • Exploits für konkrete Schwachstellen erstellen • Eigenes Vorgehen rechtlich und ethisch bewerten
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 1
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung IT-Sicherheit Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Vertiefungsrichtung Informatik in der Bildung

1	**Modulbezeichnung** 93210	**Didaktik der Informatik I** Teaching computer science I	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Marc-Pascal Berges
5	**Inhalt**	• Grundlagen der informatikbezogenen Unterrichtsplanung und -gestaltung • Informatik und Informatikdidaktik im Wissenschaftskontext • Informatische Modellbildung • Programmieren im Informatikunterricht • Werkzeuge für den Informatikunterricht • Unterrichtsmethoden und -techniken • Aufgaben und Aufgabenkultur für einen kompetenzorientierten Informatikunterricht
6	**Lernziele und Kompetenzen**	Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, Informatikunterricht begründet zu planen, durchzuführen und zu reflektieren. Sie sind in der Lage, Unterrichtsinhalte motivierend, schülernah, verständlich und zielführend zu vermitteln. Sie können Elemente der Informatik in Alltagssituationen zur Motivation und als Modellierungsgrundlage heranziehen, Realsituationen informatisch modellieren, den Prozess des Modellierens schülerbezogen gestalten und Schülerinnen und Schüler beim Modellieren unterstützen. Sie • analysieren informatische Unterrichtsgegenstände fachdidaktisch und geben Unterrichtsziele outcomeorientiert an • charakterisieren die Wissenschaft Informatik und ihre Rolle im Bildungskontext (Computer Literacy, Great Principles of Computing, Computational Thinking) und geben eine eigene Definition für Informatik an • geben Ziele des Informatikunterrichts (gemäß Lehrplan Bayern) an und beschreiben beispielhaft Möglichkeiten zur Umsetzung dieser Ziele • geben zu Inhalten des Lehrplans konkrete durch die SuS zu erwerbende Kompetenzen an und gestalten entsprechenden Unterricht • beschreiben die roten Fäden" in den Lehrplänen für Informatik in Bayern und berücksichtigen diese in der Gestaltung von Unterricht • erläutern den Informationszentrierten Ansatz und seinen Einfluss auf den bayerischen Lehrplan • ordnen Inhalte des Lehrplans dem Gesamtkonzept des Lehrplans zu • beschreiben Informatische Modellbildung, geben Beispiele und Darstellungsformen für Modellierungstechniken an und

		begründen die Relevanz informatischen Modellierens für die Schulinformatik • erläutern und illustrieren den Modellbegriff und Modellbildungsprozess aus Sicht der Informatik an selbst gewählten Beispielen • wenden Theorie und Begriffe informatischer Modellbildung in der Gestaltung und Bewertung von Unterrichtsszenarien an • ordnen Beispiele und Werkzeuge des Informatikunterrichts den Klassen von Modellen zu (EIS) • diskutieren Stellenwert, Rolle und Ziele des Programmierens in der informatischen Bildung und im informationszentrierten Ansatz • diskutieren den Stellenwert von Modellierung und Programmierung im Informatikunterricht ihrer Schulform • grenzen die Begriffe Modellieren, Programmieren und Codieren voneinander ab • begründen aus historischer und aktueller Perspektive den Einsatz von Methoden und Werkzeugen für die Vermittlung von Programmierkompetenz • diskutieren den Einsatz visueller und textueller Programmiersprachen • wenden Werkzeuge für den Informatikunterricht begründet in der Gestaltung von Unterricht an. • nennen Kriterien für Werkzeuge und wählen Werkzeuge für den Informatikunterricht begründet aus • begründen den Einsatz der Projektmethode im Informatikunterricht erläutern deren Ziele • ordnen die Projektmethode in Kategorien der Sozial- und Lehr-/Lernformen ein • erstellen ein Szenario für ein Informatikunterrichtsprojekt • vergleichen Wasserfallmodell und Agile Methoden als Grundlage für die Durchführung eines Informatikprojekts • beschreiben agile Techniken und wenden diese in der methodischen Unterrichtsgestaltung an • strukturieren und bewerten Unterrichtsmethoden für den Informatikunterricht • wählen für gegebene Inhalte und Kompetenzen adäquate Unterrichtsmethoden begründet aus • erläutern verschiedene Unterrichtstechniken und -prinzipien anhand von adressierten Problemen, Zielen und Beispielen • nennen Qualitätskriterien für Aufgaben und Leitfragen zur Aufgabenentwicklung und wenden diese in der Analyse und Entwicklung von Aufgaben an • entwickeln Aufgaben hinsichtlich eines kompetenzorientierten Informatikunterrichts unter verschiedenen Gesichtspunkten (z.B. Öffnen von Aufgaben, Kontextorientierung, Kreativität) (weiter) und ordnen diese den GI-Bildungsstandards zu
7	**Voraussetzungen für die Teilnahme**	• Modul 3050 Algorithmen und Datenstrukturen • Modul 3200 Theoretische Informatik für Lehramtsstudierende

		• Modul 3130 Konzeptionelle Modellierung
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Informatik in der Bildung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Portfolio
11	**Berechnung der Modulnote**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Hubwieser, Peter. Didaktik der Informatik. Springer-Verlag, 2007. • Schubert, Sigrid, and Andreas Schwill. Didaktik der Informatik. Spektrum Akademischer Verlag, 2011. • Werner Hartmann, Michael Näf, and Raimond Reichert. Informatikunterricht planen und durchführen. Springer, 2007. • Meyer, Hilbert. Leitfaden Unterrichtsvorbereitung. Cornelsen Scriptor, 2007.

1	**Modulbezeichnung** 93220	**Didaktik der Informatik II** Teaching computer science II	**5 ECTS**
2	Lehrveranstaltungen	Praktikum: Praktikum zur Anwendung von Informatiksystemen aus fachdidaktischer Sicht (4 SWS) Vorlesung: Didaktik der Informatik II (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Claudia Neuner Prof. Dr. Marc-Pascal Berges	

4	**Modulverantwortliche/r**	Prof. Dr. Marc-Pascal Berges
5	**Inhalt**	• Theoretische Fundierung der Didaktik der Informatik • Voraussetzungen und Rahmenbedingungen für Informatikunterricht • Lern- und Kompetenzziele des Informatikunterrichts
6	**Lernziele und Kompetenzen**	Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, Informatikunterricht zu planen, durchzuführen, zu reflektieren und auf wissenschaftlicher Grundlage weiterzuentwickeln. Sie haben vertiefte fachbezogene Reflexionskompetenzen, informatikdidaktische Basis- und diagnostische Kompetenzen sowie informatikunterrichtsbezogene Handlungskompetenzen erworben. Sie sind in der Lage, Unterrichtsinhalte motivierend, schülernah, verständlich und zielführend zu vermitteln. Hierzu können sie entscheiden, welche Inhalte der Informatik für die Schule relevant sind und diese lerngruppenadäquat aufbereiten. Sie können Elemente der Informatik in Alltagssituationen zur Motivation und als Modellierungsgrundlage heranziehen und den Beitrag des Faches zur Allgemeinbildung beschreiben und transportieren. Sie können Realsituationen informatisch modellieren, den Prozess des Modellierens schülerbezogen gestalten und Schülerinnen und Schüler beim Modellieren unterstützen.
7	**Voraussetzungen für die Teilnahme**	Es wird eine vorherige erfolgreiche Teilnahme am Modul "Didaktik der Informatik 1" (93211) wird empfohlen
8	**Einpassung in Studienverlaufsplan**	Semester: 3
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Informatik in der Bildung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Studienleistung Klausur (120 Minuten) Im Rahmen des Moduls erstellen die Studierenden ein Portfolio im Umfang von 40-100 Seiten. Sie wenden dabei die Methode des reflexiven Schreibens an.
11	**Berechnung der Modulnote**	Studienleistung (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 60 h

14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Hubwieser, Peter. Didaktik der Informatik. Springer-Verlag, 2007. • Schubert, Sigrid, and Andreas Schwill. Didaktik der Informatik. Spektrum Akademischer Verlag, 2011.

1	**Modulbezeichnung** 93149	**Praktikum Informatik in der Bildung (PIB)** Laboratory: Computer science in education (PIB)	**5 ECTS**
2	Lehrveranstaltungen	Praktikum: Praktikum Informatik in der Bildung (4 SWS)	5 ECTS
3	Lehrende	Prof. Dr. Marc-Pascal Berges	

4	**Modulverantwortliche/r**	Prof. Dr. Marc-Pascal Berges
5	**Inhalt**	Die Studierenden erstellen fachdidaktische Materialien zu aktuellen Themen aus den Schulcurricula oder der fachdidaktischen Forschung. Sie arbeiten dabei im Team. Außerdem erstellen Sie zu den erstellten Materialien eine fachdidaktische Analyse und ein Einsatzszenario. Dabei erlangen Sie ein tieferes Verständnis der zugrundliegenden Curricula.
6	**Lernziele und Kompetenzen**	Die Studierenden sind in der Lage geeignete fachdidaktische Materialien für Schule und Hochschule zu erstellen. Sie können auf der Basis vorhandener Curricula und theoretischer Grundlagen eine fachdidaktische Analyse vornehmen. Außerdem sind sie in der Lage ein mögliches Einsatzszenario auf der Basis der Grundsätze guter Unterrichtsplanung zu beschreiben.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Informatik in der Bildung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Vertiefungsrichtung Medizinische Informatik

1	**Modulbezeichnung** 44445	**Cognitive Neuroscience for AI Developers** Cognitive neuroscience for AI developers	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Cognitive Neuroscience for AI Developers (4 SWS)	5 ECTS
3	Lehrende	Prof. Dr. Andreas Kist Dr. rer. nat. Patrick Krauß Dr. rer. nat. Achim Schilling	

4	**Modulverantwortliche/r**	
5	**Inhalt**	Neuroscience has played a key role in the history of artificial intelligence (AI), and has been an inspiration for building human-like AI, i.e. to design AI systems that emulate human intelligence. Neuroscience provides a vast number of methods to decipher the representational and computational principles of biological neural networks, which can in turn be used to understand artificial neural networks and help to solve the so called black box problem. This endeavour is called neuroscience 2.0 or machine behaviour. In addition, transferring design and processing principles from biology to computer science promises novel solutions for contemporary challenges in the field of machine learning. This research direction is called neuroscience-inspired artificial intelligence. The course will cover the most important works which provide the cornerstone knowledge to understand the biological foundations of cognition and AI, and applications in the areas of AI-based modelling of brain function, neuroscience-inspired AI and reverse-engineering of artificial neural networks.
6	**Lernziele und Kompetenzen**	The students • Explain the principles of neural information processing in the brain • compare and analyze methods from neuroscience to study neural networks • explain the neuroscientific underpinnings of artificial intelligence • explain principles and concepts of cognitive science • explain principles and concepts of neuroscience • compare and analyze machine learning methods to analyze neural data • explain approaches from deep learning to model brain function • discuss the commonalities of neuroscience and artificial intelligence • implement the presented methods in Python • explain concepts from cognitive neuroscience for the design of artificial intelligence systems
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4

9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Medizinische Informatik Bachelor of Science Informatik 20222 Vertiefungsrichtung Mustererkennung Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel (90 Minuten)
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 60 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Gazzaniga, Michael. Cognitive Neuroscience - The Biology of the Mind. W. W. Norton & Company, 2018. Ward, Jamie. The Student's Guide to Cognitive Neuroscience. Taylor & Francis Ltd., 2019. Bermúdez, José Luis. Cognitive Science: An Introduction to the Science of the Mind. Cambridge University Press, 2014. Friedenberg, Jay D., and Silverman, Gordon W. Cognitive Science: An Introduction to the Study of Mind. SAGE Publications, Inc., 2015. Gerstner, Wulfram, et al. Neuronal dynamics: From single neurons to networks and models of cognition. Cambridge University Press, 2014.

1	**Modulbezeichnung** 93300	**Einführung in die Medizinische Informatik** Introduction to medical informatics	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Einführung in die Medizinische Informatik für Informatik-Nebenfachstudierende (4 SWS)	5 ECTS
3	Lehrende	PD Dr. Werner Adler Wolfgang Rödle Prof. Dr. Wolfgang Uter Dr. Meik Kunz Prof. Dr. Oliver Amft PD Dr. Heiko Gaßner Prof.Dr.med. Thomas Ganslandt	

4	**Modulverantwortliche/r**	Wolfgang Rödle
5	**Inhalt**	Die Vorlesung soll an medizinischer Informatik interessierten Studenten einen groben thematischen Überblick über verschiedene Schwerpunkte der Medizinischen Informatik geben. Vorgestellte Themenbereiche: • Informationssysteme im Gesundheitswesen • Medizinische Biometrie und Epidemiologie • mHealth • Bioinformatik
6	**Lernziele und Kompetenzen**	Ziel dieser Vorlesung ist es, den Studenten einen ersten Überblick über die verschiedenen methodischen Schwerpunkte der Medizinischen Informatik zu verschaffen, damit diese die Kompetenz erwerben, das Fach nachfolgend anhand ihrer eigenen Interessensgebiete zu vertiefen. Neben den Vorlesungen sind eine Reihe von Übungsaufgaben zu bearbeiten. Diese beinhalten einerseits das selbständige Recherchieren und Erschließen verschiedener Themen/Probleme/ Aufgabenstellungen der Medizinischen Informatik, andererseits ermöglichen sie das Vertiefen der in der Vorlesung vermittelten Lehrinhalte. Fachkompetenz Wissen • nennen die Grundbegriffe im Bereich Epidemiologie und Biometrie • nennen grundlegende Werkzeuge und Fragestellungen der Bioinformatik Verstehen • Informationssysteme und ihre Zusammenhänge im Krankenhaus erläutern • Prozessmodelle im Krankenhaus skizzieren • Grundlagen zu klinischer Dokumentation erklären • einen Überblick zu Gesundheitstelematik-Werkzeugen (z.B. eRezept oder eMedikationsplan) veranschaulichen • Funktionen und Inhalte der verschiedenen elektronischen AKten (eKA, ePA, eGA) aufzeigen

		• erläutern die Zielsetzung von mHealth und Digital Health, sowie der wesentlichen Eigenschaften von kontextbewussten Systemen • erklären den Aufbau und die grundlegenden Eigenschaften von inertialen Messsystemen und akustischen Aufnehmern • erklären empirische Methoden zur Analyse der Modellanpassung • erklären grundlegende statistische Tests • erklären grundlegende Konzepte und Algorithmen der Bioinformatik • erklären bioinformatische Methoden für die Analyse von DNA-, RNA- und Protein-Sequenzen • erklären bioinformatische Methoden für die klinische Entscheidungsunterstützung Anwenden • wenden das Bayes'sche Theorem und Varianten an, sowie die Maximum A-posterior Entscheidungsregel zur Klassifikation von Merkmalen aus Sensordaten • wenden die Grundkenntnisse auf einfache Problemstellungen in Biostatistik und Epidemiologie an • wenden grundlegende bioinformatische Analysemethoden auf einfache medizinische Fragestellungen an Analysieren • analysieren Kontexterkennung, insbesondere in der isolierten Klassifikation von Aktivitäten anhand von inertialen Sensordaten aus körpergetragenenen / Wearable Systemen
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Nebenfach Medizinische Informatik Bachelor of Science Informatik 20222 Schwerpunkt Informationssysteme im Gesundheitswesen Bachelor of Science Informatik 20222 Schwerpunkt Physiologie Bachelor of Science Informatik 20222 Vertiefungsrichtung Medizinische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich (20 Minuten)
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 22910	**Informationssysteme im Gesundheitswesen**	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Informationssysteme im Gesundheitswesen [Informatik/Medizintechnik] (4 SWS)	5 ECTS
3	Lehrende	Wolfgang Rödle Prof.Dr.med. Thomas Ganslandt PD Dr. Dennis Toddenroth Prof. Dr. Hans-Ulrich Prokosch Dr. med. Lorenz Kapsner	

4	**Modulverantwortliche/r**	Prof. Dr. Hans-Ulrich Prokosch
5	**Inhalt**	In diesem Modul wird ein Überblick über die Architektur und Einsatzgebiete von Informationssystemen im Gesundheitswesen gegeben. Es wird hierbei sowohl auf Informationssysteme im Krankenhaus als auch auf Systeme im niedergelassenen Bereich eingegangen. Aufgabengebiete und Funktionalitäten beispielhafter medizinischer Informationssysteme werden vorgestellt. Gleichzeitig wird auch auf die EDV-gestützte Vernetzung der verschiedenen Institutionen im Gesundheitswesen (Gesundheitstelematik, Telemedizin) sowie die Entstehung elektronischer Patientenakten eingegangen. Das Modul besteht aus einem Vorlesungsteil (2x 90 Minuten) und einem Teil für eigene praktische Übungen, in denen die jeweiligen Themen u.a. durch Informationsrecherche im Internet vertieft werden.
6	**Lernziele und Kompetenzen**	Die Studierenden Fachkompetenz Verstehen • erläutern unterschiedliche Prozesse des Gesundheitswesens die durch IT-Systeme unterstützt werden. • unterscheiden zwischen der Vielzahl verschiedener IT-Systeme im Gesundheitswesen (z.B. klinische Arbeitsplatzsysteme, Elektronische Krankenakte, Abteilungssysteme in diagnostischen und therapeutischen Funktionsbereichen, Arztpraxissysteme, Apothekeninformationssysteme, Anwendungen zur Forschungs-IT) und deren Funktionalität • verstehen die Heterogenität der IT-Landschaften im Gesundheitswesen und die Herausforderungen in Bezug auf die Interoperabilität zwischen den Systemen Anwenden • identifizieren Kommunikationsprozesse im Krankenhaus • erfassen der passenden Kommunikations- und Nachrichtenstandards des Gesundheitswesens zur Etablierung von Schnittstellen • analysieren IT-Landschaften in Einrichtungen des Gesundheitswesens • konzipieren eine in sich konsistente Gesamt-IT-Architektur Analysieren

		• erschließen sich Prozessabläufe im Gesundheitswesen und können daraus notwendige IT-Systeme und deren Schnittstellen ableiten
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Nebenfach Medizinische Informatik Bachelor of Science Informatik 20222 Schwerpunkt Informationssysteme im Gesundheitswesen Bachelor of Science Informatik 20222 Vertiefungsrichtung Medizinische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 22850	**Wissensbasierte Systeme in der Medizin 1** Medical knowledge processing 1	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Hans-Ulrich Prokosch
5	**Inhalt**	Die Studierenden grenzen konventionelle Software von wissensbasierten Systemen bzw. medizinischen Expertensystemen ab. erklären grundlegende Inferenzstrategien wie Vorwärts- und Rückwärtsverkettung. kennen den einzigen verbreiteten Standard für medizinische Wissensrepräsentation. nutzen die Arden-Syntax zum Erstellen von Wissensmodulen. kennen historisch bedeutsame Expertensysteme und deren Inferenzstrategien. erklären grundlegende Aspekte der Arzneimitteltherapiesicherheit. diskutieren die Schwierigkeiten bei der Integration wissensbasierter Funktionen in die klinische Routine.
6	**Lernziele und Kompetenzen**	Die Studierenden erklären den Unterschied zwischen konventioneller Software und wissensbasierten Systemen bzw. medizinischen Expertensystemen. erklären grundlegende Inferenzstrategien wie Vorwärts- und Rückwärtsverkettung. erklären und nutzen den bisher einzigen Standard für medizinische Wissensrepräsentation. kennen historisch bedeutsame Expertensysteme und deren Inferenzstrategien. erstellen selbständig standardisierte Wissensmodule. erklären grundlegende Aspekte der Arzneimitteltherapiesicherheit. verstehen die Schwierigkeiten bei der Integration wissensbasierter Funktionen in die klinische Routine.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Nebenfach Medizinische Informatik Bachelor of Science Informatik 20222 Schwerpunkt Informationssysteme im Gesundheitswesen Bachelor of Science Informatik 20222 Vertiefungsrichtung Medizinische Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich (20 Minuten)

11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Vertiefungsrichtung Kryptographie

1	**Modulbezeichnung** 93006	**Cryptocurrencies**	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Carina Köhner Prof. Dr. Dominique Schröder
5	**Inhalt**	Bitcoin was invented in 2009 and led to paradigm shifts in cryptocurrencies and distributed computing. Before its invention, two beliefs were carved in stone: First, cryptocurrencies are only of theoretical interest. Second, the Byzantine generals' problem cannot be solved in virtually any practical setting. Both views were refuted and led to new broad fields of research. This lecture introduces cryptocurrencies and explains why Bitcoin succeeded where many other approaches failed before. A preliminary list of topics covered are the following (subject to change): • Cryptographic e-cash • Blockchain • Bitcoin Protocol • Nakamoto Consensus • Bitcoin Community and Politics • Alternative Mining Approaches • Alternative Coins: Ethereum, Anonymous Payments, Zero-Knowledge Cash • Other state-of-the-art topic related to cryptocurrencies
6	**Lernziele und Kompetenzen**	Fachkompetenz Verstehen Die Studierenden kennen die wesentlichen Eigenschaften digitaler Währungen und können diese auch vergleichen. Analysieren Die Studierenden können digitale Währungen untersuchen und überprüfen, ob diese die grundlegenden Eigenschaften einer digitalen Währung erfüllen. Evaluieren (Beurteilen) Die Studierenden können unterschiedliche digitalen Währung miteinander vergleichen und je nach Anwendung einen geeigneten Kandidaten ermitteln.
7	**Voraussetzungen für die Teilnahme**	Basic knowledge in math, probability theory, cryptography, and computer security is assumed. It would be ideal if you have done a cryptography and/or security class at FAU or during your previous studies somewhere else. Motivated students without this background are also welcome to the course. However, we would expect some more spirited efforts from you.
8	**Einpassung in Studienverlaufsplan**	Semester: 4

9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Kryptographie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Introduction to Modern Cryptography, Jonathan Katz and Yehuda Lindell, CRC Press 2nd edition Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction by Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder In jeder Vorlesung werden Referenzen auf wissenschaftliche Arbeiten gegeben.

1	**Modulbezeichnung** 93206	**Cryptographic Communication Protocols** Cryptographic communication protocols	**5 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Prof. Dr. Paul Rösler
5	**Inhalt**	Almost every secure communication protocol relies on some form of initial key exchange. Using such an exchanged initial key, the communication participants can employ encryption protocols to protect the actual transmitted payload. In order to achieve strong(er) security guarantees, modern communication systems, such as secure messengers, additionally renew the exchanged key material regularly during the communication. This lecture begins with understanding the simplest forms of key exchange – Key Encapsulation Mechanisms (e.g., ElGamal KEM) and Non-Interactive Key Exchange (e.g., Diffie Hellman Key Exchange) – and gradually introduces relevant extensions to eventually understand how modern (messaging) protocols renew the exchanged key material. A preliminary list of topics covered are the following (subject to change): Simple Building Blocks: Encryption, Key Encapsulation Mechanisms, Diffie Hellman Key Exchange, and ElGamal; Standard Security Definitions: Indistinguishability, Chosen Plaintext Attacks, and Chosen Ciphertext Attacks; Definitions for Key Encapsulation Mechanisms; Forward Secrecy for Key Exchange: Two-Pass Key Exchange, Forward-Secure KEMs, and Identity Based Encryption; Authenticated Key Echange: Signed Diffie Hellman and KEM-based Authentication; Real-World Key Exchange: TLS, Noise, and X3DH; Forward Security and Post-Compromise Security: From Symmetric Channel to Ratcheted Key Exchange; Ratcheted Key Exchange: From Unidirectional to Bidirectional Communication and back; Unidirectional Ratcheted Key Exchange: Construction and Proof; Key-Updatable KEM and Real-World Ratcheting: Signal, WhatsApp, and Extensions; Double Ratchet and Ratcheted Key Exchange in Groups: Tree-based Diffie Hellman; TreeKEM and Extensions: Dynamic Groups, Concurrency, MLS Standard, and the European DMA; Summary, Questions and Answers.
6	**Lernziele und Kompetenzen**	Verstehen: Die Studierenden verstehen, wie Sicherheitsziele abstrahiert und definitiert werden, sie kennen die wesentlichen Eigenschaften moderner Schlüsselaustauschprotokolle und können diese auch vergleichen. Analysieren:

		Die Studierenden können moderne Schlüsselaustauschprotokolle untersuchen und überprüfen, ob diese die gängigen Eigenschaften erfüllen. Evaluieren (Beurteilen): Die Studierenden können für den jeweiligen Einsatzzweck geeignete Eigenschaften spezifizieren, unterschiedliche Schlüsselaustauschprotokolle miteinander vergleichen und dementspreched einen geeigneten Kandidaten ermitteln.
7	**Voraussetzungen für die Teilnahme**	Basic knowledge in math, probability theory, cryptography, and computer security is assumed. It would be ideal if you attended a cryptography and/or security class at FAU or during your previous studies somewhere else. Motivated students without this background are also welcome to the course. However, we would expect some more spirited efforts from you.
8	**Einpassung in Studienverlaufsplan**	Semester: 3
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Kryptographie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich (60 Minuten)
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch oder Englisch
16	**Literaturhinweise**	Introduction to Modern Cryptography, Second Edition, Jonathan Katz and Yehuda Lindell; A Graduate Course in Applied Cryptography, Dan Boneh and Victor Shoup. In jeder Vorlesung werden Referenzen auf wissenschaftliche Arbeiten gegeben.

1	**Modulbezeichnung** 93015	**Einführung in die moderne Kryptographie** Introduction to modern cryptography	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Einführung in die moderne Kryptographie (4 SWS) Übung: Einführung in die moderne Kryptographie (Übung) (2 SWS)	5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr. Dominique Schröder	

4	**Modulverantwortliche/r**	Carina Köhner Prof. Dr. Dominique Schröder
5	**Inhalt**	This course gives a comprehensive introduction to modern cryptography. The course also serves as a base for other courses on cryptography that are offered by the chair. The topics covered are the following: • Information theoretic security • Computational security • Private key Encryption • Message Authentication Codes • Hash functions • Public key Encryption • Digital Signatures More advanced topics may be covered if time permits.
6	**Lernziele und Kompetenzen**	On successfully passing the course, the student is guaranteed to be knowledgeable on the basic concepts of provable security.
7	**Voraussetzungen für die Teilnahme**	No previous knowledge in Cryptography or computer Security is required.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Kryptographie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 165 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Introduction to Modern Cryptography Jonathan Katz and Yehuda Lindell 2nd Edition (2014)

		(Chapman & Hall/CRC Cryptography and Network Security Series) ISBN-13: 978-1466570269

1	**Modulbezeichnung** 93680	**Introduction to Privacy** Introduction to privacy	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Dominique Schröder
5	**Inhalt**	This course gives an introduction to people-centric privacy and shows how to use methods and techniques from cryptography, distributed computing, and data analysis. It covers topics from different areas such as: • Data anonymization techniques such as k-anonymity, k-closeness, l-diversity, differential privacy • Anonymous communication • Privacy for blockchains and cryptocurrencies • Privacy enhancing technologies
6	**Lernziele und Kompetenzen**	On successfully passing the course, the students understand the difficulty of realizing privacy-preserving systems, understand the different terminologies, and use privacy-enhancing technologies to improve privacy in practical systems.
7	**Voraussetzungen für die Teilnahme**	Basic knowledge in IT security, mathematics, networks, and cryptography
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Kryptographie Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	Unregelmäßig
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Scientific papers that will be referred to during each lecture

Vertiefungsrichtung Visual Computing

1	**Modulbezeichnung** 93173	**Computational Visual Perception** Computational visual perception	**7,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Computational Visual Perception (4 SWS) Projektseminar: Tutorials to Computational Visual Perception (2 SWS)	2,5 ECTS 5 ECTS
3	Lehrende	Prof. Dr. Andreas Kist Dr. rer. nat. Patrick Krauß Prof. Dr. Tim Weyrich Prof. Dr. Bernhard Egger	

4	**Modulverantwortliche/r**	Prof. Dr. Bernhard Egger Prof. Dr. Andreas Kist Prof. Dr.-Ing. Andreas Maier
5	**Inhalt**	How do humans perceive the visual world? How can we build computational models to mimic this human perception? And how can we validate those computational models? This course is designed as an introduction to enable you to build computational models for human visual perception. It will therefore provide an introduction into the human visual system building on the course on cognitive neuroscience for AI developers. You will learn how the human eye and brain process visual input and what we currently know about the ventral visual stream. We will look at computational models for all different levels of visual processing and discuss how well they measure behavioral data. This lecture is designed to be at the intersection of Computer Science (Computer Vision and Graphics) and Cognitive Neuroscience. After an initial introductory phase, you will in small teams (1-3 students) perform a project to build prototypes for computational models for visual processing, reproduce recent scientific results or experiment with existing models. In addition to the project phase we will read and discuss recent research papers studying potential computational models and investigate how we can evaluate computational models. Please sign up via studon
6	**Lernziele und Kompetenzen**	By the end of this course, students will be able to • Describe the basic processing steps of visual input in the human brain • Build a computational model for a known processing step • Read recent papers in the discipline and design a follow-up experiment • Choose/design and conduct a small research project • Choose adequate methods to evaluate a computational model • Work in and manage projects
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4

9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Graphische Datenverarbeitung Bachelor of Science Informatik 20092 Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung schriftlich (60 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) schriftlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 43821	**Computer Graphics** Computer graphics	**5 ECTS**
2	Lehrveranstaltungen	Übung: CGTut (1 SWS) Vorlesung: Computer Graphics (3 SWS)	1,25 ECTS 3,75 ECTS
3	Lehrende	Nikolai Hofmann Linus Franke Laura Fink Prof. Dr. Marc Stamminger	

4	**Modulverantwortliche/r**	Prof. Dr. Marc Stamminger
5	**Inhalt**	Die Vorlesung gibt eine Einführung in die Computergraphik: • Graphik Pipeline • Clipping • 3D Transformationen • Hierarchische Display Strukturen • Perspektive und Projektionen • Visibilitätsbetrachtungen • Rastergraphik und Scankonvertierung • Farbmodelle • Lokale und globale Beleuchtungsmodelle • Schattierungsverfahren • Ray Tracing und Radiosity • Schatten und Texturen Contents: This lecture covers the following aspects of Computer Graphics: • graphics pipeline • clipping • 3D transformations • hierarchical display structures • perspective transformations and projections • visibility determination • raster graphics and scan conversion • color models • local and global illumination models • shading models • ray tracing and radiosity • shadows and textures
6	**Lernziele und Kompetenzen**	Die Studierenden • geben die unterschiedlichen Schritte der Graphik Pipeline wieder • erklären die Funktionsweise der Clippingalgorithmen für Linien und Polygone • beschreiben, charakterisieren und berechnen affine und perspektivische Transformationen in 3D und veranschaulichen die allgemeine Form der Transformationsmatrix in homogener Koordinaten • skizzieren die Verfahren zur Tiefe- und Visibilitätsberechnung

		• vergleichen die unterschiedlichen Farbmodelle der Computergraphik • illustrieren und untersuchen die Datenstrukturen zur Beschreibung virtueller 3D Modelle und komplexer Szenen • erläutern die Funktionsweise der Rasterisierung und Scankonvertierung in der Graphikpipeline • lösen Aufgaben zu Beleuchtung und Texturierung von 3D virtuellen Modellen • klassifizieren Schattierungsverfahren • bestimmen den Unterschied zwischen lokaler und globaler Beleuchtung und formulieren Algorithmen für Ray Tracing und Radiocity *Educational objectives and skills:* Students should be able to • describe the processing steps in the graphics pipeline • explain clipping algorithms for lines and polygons • explain, characterize and compute affine and perspective transformations in 2D and 3D, and provide an intuitive description of the general form of corresponding transformation matrices in homogeneous coordinates • depict techniques to compute depth, occlusion and visibility • compare the different color models • describe data structures to represent 3D virtual models and complex scenes • explain the algorithms for rasterization and scan conversion • solve problems with shading and texturing of 3D virtual models • classify different shadowing techniques • explain the difference between local and global illumination techniques and formulate algorithms for ray tracing and radiosity
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Graphische Datenverarbeitung Bachelor of Science Informatik 20092 Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	• P. Shirley: Fundamentals of Computer Graphics. AK Peters Ltd., 2002 • Hearn, M. P. Baker: Computer Graphics with OpenGLD. Pearson • Foley, van Dam, Feiner, Hughes: Computer Graphics - Principles and Practice • Rauber: Algorithmen der Computergraphik • Bungartz, Griebel, Zenger: Einführung in die Computergraphik • Encarnação, Strasser, Klein: Computer Graphics

1	**Modulbezeichnung** 43394	**Computergraphik (Vorlesung mit Übung und Praktikum)** Lecture with tutorial and laboratory: Computer graphics	**7,5 ECTS**
2	Lehrveranstaltungen	Übung: CGTut (1 SWS) Vorlesung: Computer Graphics (3 SWS) Übung: CGTutP (2 SWS)	1,25 ECTS 3,75 ECTS 2,5 ECTS
3	Lehrende	Linus Franke Laura Fink Prof. Dr. Marc Stamminger Nikolai Hofmann	

4	**Modulverantwortliche/r**	Prof. Dr. Marc Stamminger
5	**Inhalt**	Die Vorlesung gibt eine Einführung in die Computergraphik: • Graphik Pipeline • Clipping • 3D Transformationen • Hierarchische Display Strukturen • Perspektive und Projektionen • Visibilitätsbetrachtungen • Rastergraphik und Scankonvertierung • Farbmodelle • Lokale und globale Beleuchtungsmodelle • Schattierungsverfahren • Ray Tracing und Radiosity • Schatten und Texturen
6	**Lernziele und Kompetenzen**	Die Studierenden • geben die unterschiedlichen Schritte der Graphik Pipeline wieder • erklären die Funktionsweise der Clippingalgorithmen für Linien und Polygone • beschreiben, charakterisieren und berechnen affine und perspektivische Transformationen in 3D und veranschaulichen die allgemeine Form der Transformationsmatrix in homogener Koordinaten • skizzieren die Verfahren zur Tiefe- und Visibilitätsberechnung • vergleichen die unterschiedlichen Farbmodelle der Computergraphik • illustrieren und untersuchen die Datenstrukturen zur Beschreibung virtueller 3D Modelle und komplexer Szenen • erläutern die Funktionsweise der Rasterisierung und Scankonvertierung in der Graphikpipeline • implementieren 3D Transformationen mithilfe der Programmiersprache C++ und der graphischen Bibliothek OpenGl • Implementieren Beleuchtungsmodelle und Texturierung von virtuellen 3D Objekten mithilfe der Programmiersprachen OpenGL und GLSL

		• lösen Aufgaben zu Beleuchtung und Texturierung von 3D virtuellen Modellen • klassifizieren Schattierungsverfahren • bestimmen den Unterschied zwischen lokaler und globaler Beleuchtung und formulieren Algorithmen für Ray Tracing und Radiocity *Educational objectives and skills:* Students should be able to • describe the processing steps in the graphics pipeline • explain clipping algorithms for lines and polygons • explain, characterize and compute affine and perspective transformations in 2D and 3D, and provide an intuitive description of the general form of corresponding transformation matrices in homogeneous coordinates • depict techniques to compute depth, occlusion and visibility • compare the different color models • describe data structures to represent 3D virtual models and complex scenes • explain the algorithms for rasterization and scan conversion • solve problems with shading and texturing of 3D virtual models • classify different shadowing techniques • explain the difference between local and global illumination techniques and formulate algorithms for ray tracing and radiosity
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Graphische Datenverarbeitung Bachelor of Science Informatik 20092 Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (60 Minuten) Übungsleistung
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	• P. Shirley: Fundamentals of Computer Graphics. AK Peters Ltd., 2002

		• Hearn, M. P. Baker: Computer Graphics with OpenGLD. Pearson • Foley, van Dam, Feiner, Hughes: Computer Graphics - Principles and Practice • Rauber: Algorithmen der Computergraphik • Bungartz, Griebel, Zenger: Einführung in die Computergraphik • Encarnação, Strasser, Klein: Computer Graphics

1	**Modulbezeichnung** 796399	**Geometric Modeling** Geometric modeling	**5 ECTS**
2	Lehrveranstaltungen	Übung: GMTut (1 SWS) Vorlesung: Geometric Modeling (3 SWS)	- 5 ECTS
3	Lehrende	Dr. Roberto Grosso	

4	**Modulverantwortliche/r**	Dr. Roberto Grosso Prof. Dr. Marc Stamminger
5	**Inhalt**	Das Modul beschäftigt sich mit Methoden zur Modellierung dreidimensionaler Oberflächen. Typische Einsatzgebiete sind der rechnerunterstützte Entwurf (CAD, z.B. im Automobil- oder Flugzeugbau), die Rekonstruktion von Flächen aus Sensordaten oder die Konstruktion glatter Interpolationsflächen. Behandelt werden u.a. folgende Themen: • Polynomkurven • Bezierkurven, rationale Bezierkurven • B-Splines • Tensorproduktflächen • Bezier-Dreiecksflächen • polygonale Flächen • Subdivission-Verfahren This module is concerned with different aspects of modelling three-dimensional curves and surfaces. Typical areas of application are computer-aided design (CAD), reconstruction of surfaces from sensor data (reverse engineering) and construction of smooth interpolants. The lecture covers the following topics: • polynomial curves • Bézier curves, rational Bézier curves • B-splines • tensor product surfaces • triangular Bézier surfaces • polyhedral surfaces
6	**Lernziele und Kompetenzen**	Die Studierenden • erklären die Begriffe Polynomial-, Bezierkurven und B-Splines • klassifizieren und veranschaulichen die unterschiedlichen Auswertung- und Subdivision-Verfahren für Bezier-Kurven und B-Splines • veranschaulichen und ermitteln die Eigenschaften von Bezierkurven, rationalen Bezierkurven und B-Splines • beschreiben Tensorproduktflächen und skizzieren Auswertungsalgorithmen • erklären polygonale Flächen und Subdivision-Verfahren und veranschaulichen ihre Unterschiede und Eigenschaften • lernen gängige Datenstrukturen zur Darstellung polygonaler Flächen kennen • wenden die Verfahren der Geometrischen Modellierung an unterschiedlichen Beispiele an • berechnen Bezierkurven und B-Splines

		• führen Subdivission-Verfahren aus *Educational objectives and skills:* Students should be able to • explain the meaning of the terms Polynomial and Bezier curves and B-Splines • classify and illustrate the different evaluation and subdivision methods for Bezier curves and B-Splines • describe and establish the properties of Bezier curves, rational Bezier curves and B-Splines • describe tensor product surfaces and illustrated evaluation algorithms • explain polygonal surfaces and subdivision algorithms and depict their properties and differences • get used with common data structures to represent polygonal surfaces • apply geometric modeling algorithms to representative examples • compute Bezier curves and B-Splines • implement subdivision algorithms
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Graphische Datenverarbeitung Bachelor of Science Informatik 20092 Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	elektronische Prüfung mit MultipleChoice
11	**Berechnung der Modulnote**	elektronische Prüfung mit MultipleChoice (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• Hoschek, Lasser: Grundlagen der Geometrischen Datenverarbeitung • Farin: Kurven und Flächen im Computer Aided Geometric Design • de Boor: A Practical Guide to Splines • Bartels, Beatty, Barsky: Splines for Use in Computer Graphics and Geometric Modeling • Abramowski, Müller: Geometrisches Modellieren

1	**Modulbezeichnung** 43395	**Globale Beleuchtungsberechnung** Global illumination	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Marc Stamminger
5	**Inhalt**	Globale Beleuchtungsberechnung ist ein Kerngebiet der Computergrafik. Ziel ist die Simualation globaler Beleuchtungseffekte wie Schatten, Spiegelungen, indirektes Licht, Kaustiken etc. In der Vorlesung wird in die theoretischen Grundlagen der globalen Beleuchtungsrechnung eingeführt und es werden Raytracing-basierte Lösungsverfahren erläutert. Themen der Vorlesung sind: • Rekonstruktion und Sampling • BRDFs • Importance Sampling • Umgebungsbeleuchtung • Rendering Gleichung • Path Tracing • Irradiance Caching • Photon Mapping • ...
6	**Lernziele und Kompetenzen**	Fachkompetenz Verstehen Lernende haben ein Verständnis von Verfahren der globalen Beleuchtungsrechnung, unter anderem Monte-Carlo-Ray-Tracing, bidirectional Path-Tracing, Photon Mapping, Light Cuts, können diese in eigenen Worten wiedergeben und Beispiele anführen. Lernende können Importance Sampling für verschiedene Teilaspekte der globalen Beleuchtungsrechnung illustrieren und vergleichen und den Zusammenhang mit Multiple Importance Sampling erklären. Anwenden Lernende können verschiedene Samplingverfahren erklären für verschiedene hochdimensionale Integrationsprobleme der globalen Beleuchtungsrechnung anwenden. Analysieren Lernende können Zusammenhänge und Unterschiede von Verfahren zur globalen Beleuchtungsrechnung erkennen und Folgerungen ableiten.
7	**Voraussetzungen für die Teilnahme**	Die Übungen setzen Kenntnisse in C/C++ voraus.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Graphische Datenverarbeitung Bachelor of Science Informatik 20092 Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	Übungsleistung Variabel
11	**Berechnung der Modulnote**	Übungsleistung (0%) Variabel (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	Pharr et al.: Physically Based Rendering

1	**Modulbezeichnung** 93176	**Grundlagen empirischer Forschungsmethoden in der medialen Interaktion** Fundamentals of empirical research methods in media interaction	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Grundlagen empirischer Forschungsmethoden in der medialen Interaktion (2 SWS)	2,5 ECTS
3	Lehrende		

4	**Modulverantwortliche/r**	Prof. Dr. Marc Stamminger
5	**Inhalt**	Ziel der Vorlesung ist es, Studierende in die Lage zu versetzen, selbstständig Experimente planen und durchzuführen zu können, in denen Versuchspersonen Medien rezipieren oder mit ihnen interagieren. Zu den Experimenten gehören u.a. Hörtests, Bewertungen von Bildsyntheseverfahren und Codierartefakten und die Interaktion mit Smarten Assistenten. Ausgehend von Grundlagen der Wahrnehmung und Definition der Forschungsfrage werden Methoden eingeführt zu experimentellem Design, Stichprobenplanung, Versuchsplanung und -durchführung, Datenerhebung und -aufbereitung und Analyse der Versuchsergebnisse sowie Einbettung der Versuchsergebnisse in den wissenschaftlichen Kontext. Statistische Methoden inkl. Explorativer Datenanalyse und Visualisierung, Methoden der Varianzanalyse sowie nicht-parametrische Verfahren werde als Handwerkszeug zur Analyse und Interpretation der Versuchsergebnisse behandelt. Der Ablauf der Vorlesung folgt dem wissenschaftlichen Erkenntnisprozess und behandelt die einzelnen Schritte von der Forschungsfragen über die Genese geeigneter Hypothesen bis zu ihrer Überprüfung. Die Datenerhebung fokussiert sich dabei auf quantitative, rechnergestützte Verfahren, qualitative Methoden werden nur am Rande betrachtet. Neben der Messung psychologischer Konstrukte werden auch psychophysiologische Korrelate der Wahrnehmung, Bewertung und kognitiven Belastung der Versuchspersonen eingeführt und untersucht. • Einführung und Geschichte der Forschung in Mensch-Computer-Interfaces • Wahrnehmung und Erkennen • Forschungsfrage und experimentelles Design • Datenerhebung und Datenauswertung • Deskriptive Statistik und Visualisierung • Testen, Teststärke, Effektstärke, Power, Regression • Varianzanalytische Verfahren • Nicht-parametrische Methoden der Statistik • Anwendung von Test zur Beurteilung der Audioqualität • Grundlagen Psychophysiologie
6	**Lernziele und Kompetenzen**	Die Studierenden

		• verstehen den wissenschaftlichen Erkenntnisprozess in seinen Schritten von der Forschungsfrage über die Genese geeigneter Hypothesen bis zu ihrer Überprüfung • können Verfahren zur statistischen Testung mit parametrischen und nichtparametrischen Verfahren anwenden • verstehen Grundlagen psychologischer und psychophysiologischer Messungen und können diese durchführen und auswerten • können einfache Experimente mit Versuchspersonen planen und durchführen
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Graphische Datenverarbeitung Bachelor of Science Informatik 20092 Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel Übungsleistung
11	**Berechnung der Modulnote**	Variabel (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 299892	**Informationsvisualisierung** Information visualization	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Dr. Roberto Grosso
5	**Inhalt**	Aufgrund der rasanten Entwicklung der Informationstechnologie sind wir mit einer noch nie dagewesenen Flut an Daten konfrontiert. Informationsvisualisierung befasst sich mit der graphischen Darstellung abstrakter Daten, die keine räumliche Struktur aufweisen. Die Visualisierung abstrakter Daten nutzt visuelle Metaphern und Interaktion, um Information aus den Daten zu extrahieren. Typische Anwendungsszenarien sind die Analyse von Finanztransaktionen oder sozialen Netzwerken, Geographie, Textanalyse oder Visualisierung von Software-Quellcode. In dieser Vorlesung werden unterschiedliche Techniken vorgestellt, um verschieden Arten von Daten zu visualisieren. Insbesondere werden folgende Themen behandelt: • Graphen und Netzwerke • Dynamische Graphen • Hierarchien und Bäume • Multivariate Daten • Time-Series Daten • Textvisualisierung
6	**Lernziele und Kompetenzen**	Die Studierenden Wissen • zählen Datentypen der Informationsvisualisierung auf • nennen Techniken zur Visualisierung unterschiedlicher Datentypen der Informationsvisualisierung • beschreiben Anwendungsfällen für die unterschiedlichen Datentypen der Informationsvisualisierung Verstehen • stellen Algorithmen der Informationsvisualisierung dar und erläutern ihre Eigenschaften, Vorteile und Nachteile • illustrieren Techniken zu Auswertung und Analyse von Daten der Informationsvisualisierung • implementieren die vorgestellten Algorithmen in JavaScript Anwenden • wenden Algorithmen zur Visualisierung unterschiedlichen Daten an • erklären und charakterisieren Techniken der Informationsvisualisierung • Analysieren • klassifizieren Algorithmen zur Visualisierung multivariater Daten, Netzwerke, Hierarchien und Text und erklären ihrer Funktionsweise

		• erkunden die Effizienz der vorgestellten Algorithmen für unterschiedlichen Anwendungsmöglichkeiten Evaluieren • bewerten Anwendbarkeit und Performance spezieller Algorithmen der Informationsvisualisierung • vergleichen Methoden zur Analyse und Auswertung von Daten der Informationsvisualisierung • überprüfen die Anwendbarkeit der diskutierten Techniken für unterschiedliche, speziell ausgewählten Fälle
7	**Voraussetzungen für die Teilnahme**	Die Programmieraufgaben werden in JavaScript implementiert.
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Graphische Datenverarbeitung Bachelor of Science Informatik 20092 Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	elektronische Prüfung mit MultipleChoice (90 Minuten)
11	**Berechnung der Modulnote**	elektronische Prüfung mit MultipleChoice (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch

16	**Literaturhinweise**	Information Visualization
		• Robert Spence: Information Visualization: Design for Interaction
		• Stuart K. Card, Jock Mackinlay, Ben Shneiderman: Readings in Information Visualization – Using Vision to Think
		• Benjamin B. Bederson, Ben Shneiderman: The Craft of Information Visualization – Readings and Reflections
		• Tamara Munzner: Visualization Analysis and Design
		• Colin Ware: Information Visualization, Perception for Design (third edition)
		• Ricardo Mazza: Introduction to Information Visualization
		• Robert Spence: Information Visualization - An Introduction
		
		Networks / Graphs
		
		•
		  ° Graph Theory, Reinhard Diestel
		  ° Graphentheorie, Peter Tittmann
		  ° Graphs, Networks and Algorithms, Dieter Jungnickel

		•
		  ° Networks, 2nd Edition, Mark Newman
		  ° Graph Theory and Complex Networks: An Introduction, Maarten van Steen

1	**Modulbezeichnung** 43391	**Interaktive Computergraphik** Interactive computer graphics	**5 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Prof. Dr. Marc Stamminger
5	**Inhalt**	In dem Modul werden GPUs und dafür maßgeschneiderte Algorithmen behandelt: • Architektur von GPUs und Echtzeit-Rendering-Pipeline • Deferred Shading und Anti-Aliasing-Verfahren • Simulation von Umgebungsbeleuchtung • Verfahren zur Generierung von Schatten • Level-of-Detail-Verfahren zur Darstellung komplexer Szenen • Animation von Objekten
6	**Lernziele und Kompetenzen**	Fachkompetenz Verstehen Lernende können verschiedene in der Vorlesung behandelte Verfahren der interaktiven Computergraphik mit eigenen Worten erklären. Anwenden Lernende können Zusammenhänge zwischen den in der Vorlesung behandelten Verfahren der interaktiven Computergraphik erkennen, und Ideen auf neue Anwendungen übertragen. Analysieren Lernende können Unterschiede und Ähnlichkeiten der in der Vorlesung behandelten Verfahren erkennen und daraus neue Lösungen entwickeln.
7	**Voraussetzungen für die Teilnahme**	Die Übungen setzen Kenntnisse in C/C++ voraus.
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Graphische Datenverarbeitung Bachelor of Science Informatik 20092 Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Variabel
11	**Berechnung der Modulnote**	Übungsleistung (0%) Variabel (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch Englisch
16	**Literaturhinweise**	• Möller, Haines: "Real-Time Rendering"

1	**Modulbezeichnung** 93086	**Neural Graphics and Inverse Rendering** Neural graphics and inverse rendering	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Marc Stamminger
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	mündlich (30 Minuten) Übungsleistung
11	**Berechnung der Modulnote**	mündlich (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Visualization Analysis and Design, Tamara Munzner, 2014.

1	**Modulbezeichnung** 43385	**Physically-based Simulation in Computer Graphics** Physically-based simulation in computer graphics	**5 ECTS**
2	Lehrveranstaltungen	Übung: Tutorials to Physically-based Simulation in Computer Graphics (0 SWS) Vorlesung: Physically-based Simulation in Computer Graphics (0 SWS)	- 2,5 ECTS
3	Lehrende	Prof. Dr.-Ing. Tobias Günther	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Tobias Günther
5	**Inhalt**	Over the past decades, computer graphics became a vital component of the entertainment industry. Whether in regards to video games, animation movies, or visual effects in live action productions, computer animation brings virtual worlds to life. Thereby, physically-based simulations are required to reach the necessary degree of realism. Based on differential equations and numerical methods to solve them, this lecture will cover a series of algorithms used to implement physically-based simulations. Among others, we are concerned with: • kinematics and dynamics of motion (generalized coordinates), • numerical time integration techniques (explicit and implicit time integration), • rigid bodies (simulation, collision detection and response), • deformable objects (mass-spring-systems, finite-elements and thin shells), • grid-based fluid simulation (fractional step method), • particle-based fluid simulation (smoothed particle hydrodynamics and viscosity), • hybrid fluid simulation (fluid implicit particle FLIP, liquid-air interfaces), • adding detail to smoke, fire (vorticity confinement, wavelet turbulence), • shallow water waves and oceans This practical course consists of lectures, programming exercises, and a group programming project.
6	**Lernziele und Kompetenzen**	Students learn how to: • apply numerical time integration methods at practical examples • derive and analyze the properties of equations of motion • set appropriate boundary conditions • compare numerical solvers regarding stability, accuracy and performance • describe different techniques for rigid body, deformable, and fluid simulations • implement the algorithms in C++
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4

9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Graphische Datenverarbeitung Bachelor of Science Informatik 20092 Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Übungsleistung Variabel
11	**Berechnung der Modulnote**	Übungsleistung (0%) Variabel (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 43722	**Scientific Visualization** Scientific visualization	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Tobias Günther
5	**Inhalt**	The amount of data, generated in the pursuit of scientific discovery, keeps rapidly increasing across all major scientific disciplines. How can we make sense of large, time-dependent, high-dimensional and multi-variate data? This lecture provides an introduction into scientific visualization. Throughout the course, we cover the fundamental perception basics needed to convey information accurately. After categorizing different data types based on their dimensionality, we dive deeper into specific techniques for scalar, vector and tensor valued data. The lecture covers the following topics: • visualization design basics (data abstraction, visual encoding of information), • a review of scalar and vector calculus (differential properties, extremal and critical points), • data structures and data acquisition techniques (grids, interpolation, and differentiation), • indirect volume visualization (marching cubes and contour trees), • direct volume visualization (ray marching and Monte Carlo rendering), • elementary and line-based flow visualization (numerical integration, seeding, rendering), • surface-based flow visualization (integration, selection, rendering), • topology-based flow visualization (topological skeleton, bifurcations, feature flow fields), • feature-based flow visualization (vortices, material boundaries, Lagrangian coherent structures), • advanced methods (tensor visualization, uncertainty, ensembles)
6	**Lernziele und Kompetenzen**	Students are able to: • use perception basics to select appropriate visualization methods • classify data and select appropriate visualization techniques • calculate differential properties of scalar and vector fields • identify features in scalar and vector-valued data • implement numerical extraction algorithms • learn the advantages and disadvantages of common visualization techniques
7	**Voraussetzungen für die Teilnahme**	Keine

8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Graphische Datenverarbeitung Bachelor of Science Informatik 20092 Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel Electronic exam in presence with multiple choice questions (90 minutes)
11	**Berechnung der Modulnote**	Variabel (100%) The final grade of the module is determined by the exam. Exercise bonus: • Obtaining more than 80% of the points across all theoretical exercises awards an exam bonus of a third grade.
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 44481	**Visual Computing in Medicine** Visual computing in medicine	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Visual Computing in Medicine 1 (2 SWS) Vorlesung: Visual Computing in Medicine 2 (0 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	PD Dr. Peter Hastreiter PD Dr. Thomas Wittenberg	

4	**Modulverantwortliche/r**	PD Dr. Thomas Wittenberg
5	**Inhalt**	Die Flut und Komplexität medizinischer Bilddaten sowie die klinischen Anforderungen an Genauigkeit und Effizienz erfordern leistungsfähige wie auch robuste Konzepte der medizinischen Datenverarbeitung. Auf Grund der Vielfalt an Bildinformation und ihrer klinischen Relevanz spielt der Übergang von der Messung medizinischer Bilddaten (u.a. MRT, CT, PET) hin zur Analyse der Bildinhalte eine wichtige Rolle. Durch die visuelle Wiedergabe der abstrakten Daten können sowohl technische als auch medizinische Aspekte anschaulich und intuitiv verstanden werden. Aufbauend auf einem Regelkreis zur Verarbeitung medizinischer Bilddaten werden im ersten Teil (Visual Computing in Medicine I) die Eigenschaften medizinischer Bilddaten sowie grundlegende Methoden und Verfahren der medizinischen Bildanalyse und Visualisierung im Zusammenhang vermittelt. Beispiele aus der Praxis erläutern den Bezug zur medizinischen Anwendung. Darauf aufbauend werden im zweiten Teil (Visual Computing in Medicine II) konkrete Lösungsansätze für die Diagnose und Therapieplanung komplexer Krankheitsbilder erläutert. Es wird gezeigt, wie grundlegende Methoden ausgewählt und zu praktisch anwendbaren Gesamtkonzepten zusammengefasst werden. An Beispielen wird der Bezug zu Strategien und Anforderungen in der industriellen Entwicklung und klinischen Anwendung hergestellt. Ergänzend werden komplexe Methoden der medizinischen Bildanalyse und Visualisierung ausführlich besprochen. The flood and complexity of medical image data as well as the clinical need for accuracy and efficiency require powerful and robust concepts of medical data processing. Due to the diversity of image information and their clinical relevance the transition from imaging to medical analysis and interpretation plays an important role. The visual representation of abstract data allows understanding both technical and medical aspects in a comprehensive and intuitive way. Based on a processing pipeline for medical image data an overview of the characteristics of medical image data as well as fundamental methods and procedures for medical image analysis and visualization is given. Examples of clinical practice show the relation to the medical application. Based on VCMed1 the lecture VCMed2 discusses practical approaches for the diagnosis and therapy planning of complex diseases. It will be shown how fundamental methods are selected and integrated to practically applicable concepts. Examples demonstrate the relation to strategies and requirements in clinical practice and the industrial

		development process. Additionally, complex methods of medical image analysis and visualization will be explained.
6	**Lernziele und Kompetenzen**	*Visual Computing in Medicine I* Die Studierenden • erhalten einen Überblick zu Grundlagen und Unterschie-den medizinischer Bildgebungsverfahren • erwerben fundierte Kenntnisse über Gitterstrukturen, Da-tentypen und Formate medizinischer Bilddaten • üben an Beispielen die Erkennung und Interpretation un-terschiedlicher Bilddaten • erwerben Kenntnisse zu Verfahren der Vorverarbeitung, Filterung und Interpolation medizinischer Bilddaten sowie zu grundlegenden Ansätzen der Segmentierung • erlernen Prinzipien und Methoden der expliziten und im-pliziten Bildregistrierung und erhalten einen Überblick zu wichtigen Verfahren der starren Registrierung • erwerben fundierte Kenntnisse zu allen Aspekten der me-dizinischen Visualisierung (2D, 3D, 4D) von Skalar-, Vek-tor-, Tensordaten • erhalten an einfachen Beispielen einen ersten Eindruck, wie sich Visualisierung zur Steuerung von Bildanalysever-fahren und für die medizinische Diagnostik einsetzen lässt The students • get an overview of the basic principles and differences of medical imaging methods, • acquire profound knowledge about grid structures, data types and formats of medical image data, • use sample data to recognize and interpret different image data, • acquire knowledge about methods of preprocessing, filtering and interpolation of medical image data as well as on basic approaches of segmentation, • learn the principles and methods of explicit and implicit image registration and get an overview of important procedures of rigid registration, • acquire profound knowledge about all aspects of medical visualization (2D, 3D, 4D) of scalar, vector, tensor data, • get a first impression of how visualization can be used to control image analysis and medical diagnostics. *Visual Computing in Medicine II* Die Studierenden • erwerben aus Sicht der medizinischen Anwendung und konkreter Lösungsstrategien einen Einblick in komplexe Ansätze zur Bearbeitung wichtiger Krankheitsbilder • lernen die Anforderungen an und die Verknüpfung von Methoden der medizinischen Bildanalyse und Visualisie-rung zur Bearbeitung kardiologischer, neurologischer, on-kologischer und strahlentherapeutischer Fragestellungen

		• erhalten einen Überblick zu komplexen Krankheitsbildern als Grundlage für effektive und effiziente Lösungen • erwerben erweiterte Kenntnisse zur multimodalen Bildregistrierung mit nichtstarren Transformationen • erhalten vertieftes Wissen zu komplexen und aktuellen Themen der medizinischen Visualisierung (u.a. Integrationsverfahren, Transferfunktionen, Beschleunigungstech-niken mit Grafikhardware) The students • gain an insight into complex approaches to the treatment of important disease patterns from the point of view of medical application and specific solution strategies • learn the requirements and the linking of methods of medical image analysis and visualization for the processing of cardiological, neurological, oncological and radiotherapeutic questions • get an overview of complex disease pictures as a basis for effective and efficient solutions • acquire advanced knowledge to process multimodal image data using advanced methods • receive in-depth knowledge on complex and up-to-date topics of medical visualization (including integration procedures, transfer functions, acceleration techniques with graphics hardware)
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Graphische Datenverarbeitung Bachelor of Science Informatik 20092 Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	2 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	• B. Preim, C. Botha: Visual Computing for Medicine, Morgan Kaufmann Verlag, 2013 • B. Preim, D. Bartz: Visualization in Medicine - Theory, Algorithms, and Applications, Morgan Kaufmann Verlag, 2007

		• H. Handels: Medizinische Bildverarbeitung, Bildanalyse, Mustererkennung und Visualisierung für die computergestützte ärztliche Diagnostik und Therapie, Vieweg und Teubner Verlag, 2009 • P.M. Schlag, S. Eulenstein, Th. Lange: Computerassistierte Chirurgie, Elsevier Verlag, 2010 • E. Neri, D. Caramella, C. Bartolozzi: Image Processing in Radiology, Springer Verlag, 2008

1	**Modulbezeichnung** 93175	**Visualization**	**5 ECTS**
2	Lehrveranstaltungen	Übung: Tutorials to Visualization (2 SWS) Vorlesung: Visualization (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr.-Ing. Tobias Günther Xingze Tian	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Tobias Günther
5	**Inhalt**	An old English adage says "a picture is worth a 1,000 words", meaning that complex ideas are often easier to convey visually. This lecture is about the craft of creating informative images from data. Starting from the basics of the human visual perception, we will learn how visualizations are designed for explorative, communicative or confirmative purposes. We will see how data can be classified, allowing us to develop algorithms that apply to a wide range of application domains. The lecture covers the following topics: • data abstraction (data types, data set types, attribute types), • perception and mapping (marks and channels, effectiveness, pre- attentive vision, color maps), • task abstraction and validation (actions and targets), • information visualization tools (HTML, CSS, JavaScript, React, D3), • information visualization methods (tabular data, networks, trees), • scientific visualization methods (volume rendering and particle visualization), • scientific visualization tools (VTK, ParaView), • view manipulation (navigation, selection, multiple views), • data reduction (filtering, agreggation, focus and context), • lies in visualization (human biases and rules of thumb), • applications (deep learning, medical visualization, optimization) The lecture is accompanied by exercises. Theoretical exercises concentrate on the classification of data and the design and analysis of visualizations, while programming exercises using web-based technologies give examples of their implementation.
6	**Lernziele und Kompetenzen**	Students learn to: • use perceptual basics to select appropriate visualization methods • explain the steps of the visualization pipeline • calculate direct and indirect volume visualizations to given data • explain and apply interaction concepts • perform a data and requirement analysis for a given problem • explain visualization techniques for scientific and abstract data
7	**Voraussetzungen für die Teilnahme**	Keine

8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Vertiefungsrichtung Graphische Datenverarbeitung Bachelor of Science Informatik 20092 Vertiefungsrichtung Visual Computing Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Englisch
16	**Literaturhinweise**	Visualization Analysis and Design, Tamara Munzner, 2014.

Nebenfach Frankoromanistik für Informatikstudierende

1	**Modulbezeichnung** 72203	**Basismodul Einführung in die Französische Literaturwissenschaft** Introduction to French literary studies	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Sabine Friedrich
5	**Inhalt**	Vermittlung frankoromanistischen Basiswissens: • Klärung grundlegender literaturwissenschaftlicher Begriffe; • Einführung in Theorien und Methoden der Literaturwissenschaft; • Einführung in die Analyse lyrischer, narrativer und dramatischer Texte; • Entstehung und Entwicklung der Romanistik
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben fundierte Kenntnisse über Grundfragen der Frankoromanistik, • sind in der Lage, gesprochene und geschriebene Texte der studierten Sprache mit Hilfe der vermittelten Theorien und Methoden zu analysieren, • kennen grundlegende literaturwissenschaftliche Methoden und Arbeitsweisen, • können die Struktur literarischer Texte unterschiedlicher Gattungen angemessen erfassen und analysieren, • haben ein Basiswissen über die Entwicklung zentraler literaturwissenschaftlicher Konzepte, • erwerben die Fähigkeit, wissenschaftliche Positionen auszutauschen und im Gespräch zu verteidigen.
7	**Voraussetzungen für die Teilnahme**	Dringend empfohlen: Französischkenntnisse im Umfang von mindestens drei Schuljahren oder Kenntnisse entsprechend Niveau GER B1
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Frankoromanistik für Informatikstudierende Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Reader mit theoretischen Texten und Primärliteratur

1	**Modulbezeichnung** 72204	**Basismodul Einführung in die Französische Sprachwissenschaft** Basic module: Introduction to French linguistics	**5 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Prof. Dr. Silke Jansen
5	**Inhalt**	Vermittlung frankoromanistischen Basiswissens: • Stellung der romanischen Sprachen in der Welt; • Klärung grundlegender sprachwissenschaftlicher Begriffe; • Erläuterung wichtiger Sprach- und Kommunikationsmodelle; • Einführung in Theorien und Methoden verschiedener Teildisziplinen der Sprachwissenschaft jeweils unter Berücksichtigung ihrer historischen Entwicklung; • Entstehung und Entwicklung der Romanistik
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben fundierte Kenntnisse über Grundfragen der Frankoromanistik, • sind in der Lage, gesprochene und geschriebene Texte der studierten Sprache mit Hilfe der vermittelten Theorien und Methoden zu analysieren, • kennen grundlegende sprachwissenschaftliche Methoden und Arbeitsweisen, • haben ein Basiswissen über die Entwicklung zentraler sprachwissenschaftlicher Konzepte, • erwerben die Fähigkeit, wissenschaftliche Positionen auszutauschen und im Gespräch zu verteidigen.
7	**Voraussetzungen für die Teilnahme**	Dringend empfohlen: Französischkenntnisse im Umfang von mindestens drei Schuljahren oder Kenntnisse entsprechend Niveau GER B1
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Frankoromanistik für Informatikstudierende Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Stein, Achim (32010): Einführung in die französische Sprachwissenschaft. (Sammlung Metzler 307). Stuttgart: Metzler. Geckeler, Horst/Dietrich, Wolf (42007): Einführung in die französische Sprachwissenschaft. Ein Lehr- und Arbeitsbuch. Berlin: Erich Schmidt. Weitere Literaturhinweise erfolgen im Rahmen der Lehrveranstaltung

1	**Modulbezeichnung** 72213	**Basismodul Französische Sprachpraxis 1** Basic module: French language practice 1	**10 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Dr. Corina Petersilka
5	**Inhalt**	• Systematische Wiederholung und Vertiefung der französischen Grammatik (groupe nominal) • Erarbeitung eines grundlegenden, an landeskundlichen Themen orientierten Grund- und Aufbauwortschatzes unter Berücksichtigung morphologischer und semantische
6	**Lernziele und Kompetenzen**	Die Studierenden • üben den Umgang mit authentischen Dokumenten mit kulturwissenschaftlicher Relevanz, • erarbeiten die wichtigsten Strukturen der Nominalphrase im modernen Französisch, • erwerben kontextgebundenen, für landeskundliche Themen relevanten Wortschatz, • entwickeln Techniken und Methoden des Wortschatzerwerbs zur Aneignung autonomer Lernstrategien, • erlernen das aktive Beherrschen von Sprechsituationen (dialogisches und monologisches Sprechen), • üben Kurzvorträge und Diskussionen zu landeskundlich relevanten Themen.
7	**Voraussetzungen für die Teilnahme**	Empfohlen werden Französischkenntnisse auf dem Niveau von drei aufsteigenden Schuljahren oder GER: B1. Über das Niveau der tatsächlichen vorhandenen Sprachkenntnisse orientiert ein Einstufungstest zu Semesterbeginn.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Frankoromanistik für Informatikstudierende Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 210 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch

16	**Literaturhinweise**	• Klein, Hans-Wilhelm; Kleineidam, Hartmut: Grammatik des heutigen Französisch. Stuttgart: Klett, 1994. • Fischer, Wolfgang; Le Plouhinec, Anne-Marie: Thematischer Grund- und Aufbauwortschatz Französisch. Stuttgart: Klett, 2009. • Piquet, Claire

1	**Modulbezeichnung** 72224	**Basismodul Französische Sprachpraxis 2** Basic module: French language practice 2	**10 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt! Anwesenheitspflicht in Parcours III und Phonétique pratique	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Dr. Corina Petersilka
5	**Inhalt**	• Systematische Wiederholung und Vertiefung der französischen Grammatik (groupe verbal) • Ausspracheschulung zur Lautung und Intonation mit praktischen Übungen im Sprachlabor
6	**Lernziele und Kompetenzen**	Die Studierenden • wenden die wichtigsten Strukturen der Verbalphrase im modernen Französisch an, • erkennen und analysieren typische Fehlerquellen, • trainieren eine möglichst akzentfreie Aussprache des Französischen (Phonetik).
7	**Voraussetzungen für die Teilnahme**	Dringend empfohlen ist der erfolgreiche Abschluss des Basismoduls französische Sprachpraxis 1
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Frankoromanistik für Informatikstudierende Bachelor of Science Informatik 20222 - BA Frankoromanistik (Erstfach/Zweitfach): 1 von 4 Basismodulen - LaG/LaR Französisch: 1 von 5 Basismodulen
10	**Studien- und Prüfungsleistungen**	Praktische Prüfung/Test (90 Minuten) Klausur (90 Minuten) Praktische Prüfung/Test (15 Minuten) - Grammatikklausur 90‘ - Aussprachetest ca. 15‘ Wiederholbarkeit: zweimal
11	**Berechnung der Modulnote**	Praktische Prüfung/Test (40%) Klausur (40%) Praktische Prüfung/Test (20%) - Grammatikklausur/mündl. Prüfung 100 % der Modulnote - Aussprachetest/mündl. Prüfung 0 % der Modulnote
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch Französisch
16	**Literaturhinweise**	• Fischer, Wolfgang; Le Plouhinec, Anne-Marie: Thematischer Grund- und Aufbauwortschatz Französisch. Stuttgart: Klett, 2009. • Klein, Hans-Wilhelm & Kleineidam, Hartmut. Grammatik des heutigen Französisch. Stuttgart: Klett, 1994. • Bescherelle: L'Art de conjuguer. Diesterweg 2000. • Léon, Monique; Léon, Pierre: La prononciation du français B1/B2, Didier 2017.

Nebenfach Iberoromanistik für Informatikstudierende

1	**Modulbezeichnung** 72634	**Basismodul Spanische Literaturwissenschaft** Basic module: Spanish literature	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Sabine Friedrich
5	**Inhalt**	Vermittlung iberoromanistischen Basiswissens: • Klärung grundlegender literaturwissenschaftlicher Begriffe; • Einführung in Theorien und Methoden der Literaturwissenschaft; • Einführung in die Analyse lyrischer, narrativer und dramatischer Texte; • Entstehung und Entwicklung der Romanistik
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben fundierte Kenntnisse über Grundfragen der Iberoromanistik, • sind in der Lage, gesprochene und geschriebene Texte des Spanischen mit Hilfe der vermittelten Theorien und Methoden zu analysieren, • kennen grundlegende sprach- und literaturwissenschaftliche Methoden und Arbeitsweisen, • können die Struktur literarischer Texte unterschiedlicher Gattungen angemessen erfassen und analysieren, • haben ein Basiswissen über die Entwicklung zentraler literaturwissenschaftlicher Konzepte, • erwerben die Fähigkeit, wissenschaftliche Positionen auszutauschen und im Gespräch zu verteidigen.
7	**Voraussetzungen für die Teilnahme**	Dringend empfohlen: spanische Sprachkenntnisse im Umfang von ca. 120 Unterrichtsstunden, Kenntnisse entsprechend Niveau GER A2
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Iberoromanistik für Informatikstudierende Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch

16	**Literaturhinweise**	Ein Reader mit theoretischen Texten und Primärliteratur wird kursbegleitend zur Verfügung gestellt.

1	**Modulbezeichnung** 72633	**Basismodul Spanische Sprachwissenschaft** Basic module: Spanish linguistics	**5 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	Prof. Dr. Silke Jansen
5	**Inhalt**	Vermittlung iberoromanistischen Basiswissens: • Stellung der romanischen Sprachen in der Welt; • Klärung grundlegender sprachwissenschaftlichen Begriffe; • Erläuterung wichtiger Sprach- und Kommunikationsmodelle; • Einführung in Theorien und Methoden verschiedener Teildisziplinen der Sprachwissenschaft jeweils unter Berücksichtigung ihrer historischen Entwicklung; • Entstehung und Entwicklung der Romanistik
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben fundierte Kenntnisse über Grundfragen der Iberoromanistik, • sind in der Lage, gesprochene und geschriebene Texte des Spanischen mit Hilfe der vermittelten Theorien und Methoden zu analysieren, • kennen grundlegende sprachwissenschaftliche Methoden und Arbeitsweisen, • haben ein Basiswissen über die Entwicklung zentraler sprachwissenschaftlicher Konzepte, • erwerben die Fähigkeit, wissenschaftliche Positionen auszutauschen und im Gespräch zu verteidigen.
7	**Voraussetzungen für die Teilnahme**	Dringend empfohlen: spanische Sprachkenntnisse im Umfang von ca. 120 Unterrichtsstunden, Kenntnisse entsprechend Niveau GER A2
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Iberoromanistik für Informatikstudierende Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch

16	**Literaturhinweise**	Kabatek, Johannes/Pusch, Claus, D. (2009): Spanische Sprachwissenschaft. Eine Einführung (bachelor-wissen). Tübingen: Narr Francke Attempto. Weitere Literaturhinweise erfolgen im Rahmen der Lehrveranstaltung.

1	**Modulbezeichnung** 72372	**Spanische Sprachpraxis 1** Spanish language practice 1	**10 ECTS**
2	Lehrveranstaltungen	Zu diesem Modul sind keine Lehrveranstaltungen oder Lehrveranstaltungsgruppen hinterlegt!	
3	Lehrende	Zu diesem Modul sind keine Lehrveranstaltungen und somit auch keine Lehrenden hinterlegt!	

4	**Modulverantwortliche/r**	PD Dr. Joachim Christl
5	**Inhalt**	• Grundstrukturen des Spanischen, insbes.: Verbformen des Futurs und Konditionals und ihre Verwendung, Subjuntivo, Festigung des Gebrauchs der Vergangenheitstempora, Passiv, Indirekte Rede • Erweiterung des Wortschatzes anhand landeskundlicher Themen
6	**Lernziele und Kompetenzen**	Die Studierenden • können sich in Standardsituationen des privaten und beruflichen Alltags mündlich und schriftlich verständigen, • können die vier Fertigkeiten Hören, Lesen, Sprechen, Schreiben in angemessener Weise einsetzen, • kennen wichtige Unterschiede in Gepflogenheiten und Verhaltensweisen zwischen der Ausgangs- und Zielkultur und gehen reflektiert mit diesen Unterschieden um, • verstehen die besonderen Merkmale gesprochener Sprache (Intonation, redesteuernde Partikeln, persönliche Anrede u.a.) und ihrer Textsorten (Dialog, Interview, Referat usw.) und wenden diese an; Themen und Situationen des Alltags und aus dem Bereich der Studieninhalte.
7	**Voraussetzungen für die Teilnahme**	Empfohlen werden Sprachkenntnisse entsprechend Niveau GER** A2 bzw. Einführungskurs im Umfang von ca. 120 Std. Über das Niveau der tatsächlich vorhandenen Sprachkenntnisse orientiert ein Einstufungstest zu Semesterbeginn.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Iberoromanistik für Informatikstudierende Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Regelmäßige Teilnahme Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Regelmäßige Teilnahme (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 120 h Eigenstudium: 180 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch

16	**Literaturhinweise**	• Ü Español intermedio I: Aula 3 International (Verlag Klett/ Difusión) • Ü Cultura y comunicación oral: Aufnahmen von Beispielen gesprochener Sprache (verschiedene Textsorten und Sprecher aus unterschiedlichen Ländern des hispanophonen Sprachraums)<b

Nebenfach Italoromanistik für Informatikstudierende

1	**Modulbezeichnung** 72624	**Basismodul Italienische Literaturwissenschaft** Italian literary studies	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Christian Rivoletti
5	**Inhalt**	Vermittlung italoromanistisch-literaturwissenschaftlichen Basiswissens: • Klärung grundlegender literaturwissenschaftlicher Begriffe • Einführung in Theorien und Methoden der Literaturwissenschaft • Einführung in die Analyse lyrischer, narrativer und dramatischer Texte • Entstehung und Entwicklung der Romanistik
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben fundierte Kenntnisse über Grundfragen der italoromanistischen Literaturwissenschaft; • sind in der Lage, italienische gesprochene und geschriebene Texte mit Hilfe der vermittelten literaturwissenschaftlichen Theorien und Methoden zu analysieren; • kennen grundlegende literaturwissenschaftliche Methoden und Arbeitsweisen; • können die Struktur literarischer Texte unterschiedlicher Gattungen angemessen erfassen und analysieren; • haben ein Basiswissen über die Entwicklung zentraler literaturwissenschaftlicher Konzepte; • erwerben die Fähigkeit, sich (literatur)wissenschaftliche Positionen zu erschließen und im Gespräch zu explizieren.
7	**Voraussetzungen für die Teilnahme**	dringend empfohlen: Sprachkenntnisse entsprechend GER B1
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Italoromanistik für Informatikstudierende Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%) Klausurnote = Modulnote
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Italienisch

16	**Literaturhinweise**	Ein Reader mit theoretischen Texten und Primärliteratur wird kursbegleitend zur Verfügung gestellt.

1	**Modulbezeichnung** 72292	**Basismodul Italienische Sprachpraxis 1** Italian language practice 1	**10 ECTS**
2	Lehrveranstaltungen	Übung: Italiano Intermedio I (6 SWS) Übung: Comprensione e produzione orale I (2 SWS)	8 ECTS 2 ECTS
3	Lehrende	Carla Ferrara Paola Cesaroni-Meinzolt	

4	**Modulverantwortliche/r**	Paola Cesaroni-Meinzolt
5	**Inhalt**	• Grundstrukturen des Italienischen, insbesondere Morphologie, Verbformen des Indikativs, Konjunktivs und Konditionals und ihre Verwendung, Aufbau des Wortschatzes im Bereich des Alltags, des eigenen Studiums und der persönlichen Interessen; • Festigung der Grundregeln der italienischen Aussprache; • Auseinandersetzung mit ausgewählten Aspekten der italienischen Gesellschaft im sprachlichen, sozio-politischen und kulturellen Bereich.
6	**Lernziele und Kompetenzen**	Die Studierenden können • sich in Anlehnung an die Deskriptoren des GER Niveau B1 zu Themen von allgemeinem Interesse bzw. des eigenen Lebens- und Studienumfelds schriftlich und mündlich äußern bzw. austauschen; • die Grundkompetenzen Hörverstehen und Sprechen durch kollaboratives Lernen und unter Verwendung von ersten einfachen Mediationsaufgaben aufbauen; • über sprachliche Phänomene selbstständig und unter Verwendung von induktiven Verfahren reflektieren.
7	**Voraussetzungen für die Teilnahme**	Empfohlen werden Sprachkenntnisse entsprechend GER A2. Über das Niveau der tatsächlich vorhandenen Sprachkenntnisse orientiert ein Einstufungstest zu Semesterbeginn.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Italoromanistik für Informatikstudierende Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten) Abschlussklausur (120 Min.) WiSe 2022/23: alternativ schriftliche Online-Prüfung mit Authentifizierung und Kontrolle (120 Min.) Die **konkrete Prüfungsform wird spätestens vier Wochen vor der Prüfung verbindlich** kommuniziert.

11	**Berechnung der Modulnote**	Klausur (100%) Klausurnote = Modulnote
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 120 h Eigenstudium: 180 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Italienisch
16	**Literaturhinweise**	Literaturhinweise erfolgen im Rahmen der Lehrveranstaltung.

1	**Modulbezeichnung** 72623	**Basismodul Italienische Sprachwissenschaft** Italian language studies	**5 ECTS**
2	Lehrveranstaltungen	Seminar: Basisseminar italienische Sprachwissenschaft (Basismodul 72623) (2 SWS)	5 ECTS
3	Lehrende	Dr. Robert Hesselbach	

4	**Modulverantwortliche/r**	Prof. Dr. Ludwig Fesenmeier
5	**Inhalt**	Vermittlung italoromanistisch-sprachwissenschaftlichen Basiswissens: • Stellung der romanischen Sprachen in der Welt • Klärung grundlegender sprachwissenschaftlicher Begriffe • Erläuterung wichtiger Sprach- und Kommunikationsmodelle • Einführung in Theorien und Methoden verschiedener Teildisziplinen der Sprachwissenschaft unter Berücksichtigung ihrer jeweiligen historischen Entwicklung • Entstehung und Entwicklung der Romanistik
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben einen fundierten Überblick über die sprachsystematischen Beschreibungsebenen (Phonetik/ Phonologie, Morphologie, Syntax, Semantik) im Bereich der italoromanistischen Sprachwissenschaft; • sind in der Lage, italienische gesprochene und geschriebene Texte mit Hilfe der vermittelten sprachwissenschaftlichen Theorien und Methoden zu analysieren; • kennen grundlegende sprachwissenschaftliche Methoden und Arbeitsweisen; • verfügen über Basiswissen hinsichtlich der Entwicklung zentraler sprachwissenschaftlicher Konzepte; • erwerben die Fähigkeit, sich (sprach)wissenschaftliche Positionen zu erschließen und im Gespräch zu explizieren.
7	**Voraussetzungen für die Teilnahme**	Dringend empfohlen: Sprachkenntnisse entsprechend GER A2.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Nebenfach Italoromanistik für Informatikstudierende Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%) Klausurnote = Modulnote
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 120 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch Italienisch

16	**Literaturhinweise**	• Blasco Ferrer, Eduardo (1994): Handbuch der italienischen Sprachwissenschaft, Berlin: Schmidt. • Sobrero, Alberto A./Miglietta, Annarita (2021): Introduzione alla linguistica italiana. Nuova edizione riveduta e aggiornata, Roma/Bari: Laterza. • Michel, Andreas (2016): Einführung in die italienische Sprachwissenschaft, Berlin/New York: De Gruyter. • Gabriel, Christoph/Meisenburg, Trudel (2021): Romanische Sprachwissenschaft, Paderborn: Brill/Fink. Weitere Literaturhinweise erfolgen im Rahmen der Lehrveranstaltung.

Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik

1	**Modulbezeichnung** 97247	**Fertigungsmesstechnik I** Manufacturing metrology I	**5 ECTS**
2	Lehrveranstaltungen	Übung: Fertigungsmesstechnik I - Übung (2 SWS) Vorlesung mit Übung: Fertigungsmesstechnik I (2 SWS)	- -
3	Lehrende	Prof. Dr.-Ing. Tino Hausotte	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Tino Hausotte
5	**Inhalt**	• Grundlagen: Teilgebiete der industriellen Messtechnik, Grundaufgaben der Fertigungsmesstechnik, Messbedingungen und Zeitpunkte, Methoden und Teilaufgaben der Fertigungsmesstechnik, Ziele der Fertigungsmesstechnik; Begriffsdefinitionen: Messen, Überwachen, Prüfen, Überwachen, Lehren, Geschichte der Fertigungsmesstechnik, Ausrüstung in der Fertigungsmesstechnik, Grundeinteilung der Mess- und Prüfmittel, klassische Fertigungsmesstechnik, Koordinatenmesstechnik; Begriffe der Messtechnik (Wiederholung aus Grundlagenvorlesung): Messgröße, Größenwert, Messergebnis, Messwert, Messprinzip, Messmethode, Messverfahren, Empfindlichkeit, Messbereich, Auflösung (Orts- bzw. Skalenauflösung vs. Strukturauflösung, Amplituden-Wellenlängen-Diagramm), wahrer Wert, vereinbarter Wert, systematische und zufällige Messabweichung, Kalibrierung, Verifizierung, Eichung, Validierung, Messpräzision, Messgenauigkeit, Messrichtigkeit, Messunsicherheit • Längenmesstechnik (Handmessmittel und Normale): Aufgaben und Einsatz der Längenmesstechnik, Messschieber (Aufbau, Ablesung), Nonius, Parallaxenabweichung, Abweichung 1.-Ordnung, Abbe´sches Komparatorprinzip, Messvarianten mit Messschiebern, Bauformen von Messschiebern, Messschrauben (Aufbau, Ablesung), Abweichung 2.-Ordnung, Bauformen von Messschrauben, Messuhr, Feinzeiger, Fühlhebelmessgerät, induktive Messtaster (Aufbau, Kennlinie), Ursachen von Messabweichungen: Messkreis, Temperatureinflüsse, Ausdehnungskompensation, Flächenpressung und Abplattung, Deformation von Messplatten und langen Teilen, Kippungs- und Führungsabweichungen, Formabweichungen und -änderungen (Gleichdick bzw. Reuleaux-Polygone), Ellipse und Dreibogengleichdick, Dreipunktmessung, Zentrierfehler und Zentrierhilfen; Werkstoffe für Messkreise: Aluminium, Stahl, Invar 36, Super Invar 32-5, Naturstein, Polymerbeton, Keramiken, Gesintertes Siliziumcarbid, NEXCERA N113G, Titanium-Silikatglas ULE, Zerodur, mechanische Spannungen und Kriechen; Maßverkörperungen: Parallelendmaße, Fühlerlehren, Grenzrachenlehren

- Längenmesstechnik (Maßstäbe und Encoder): Maßstäbe mit visueller Ablesung: Maßstäbe mit Skalen, Auflösungsvermögen des Auges, Spiralokular, Abweichung 1.- und 2.-Ordnung (Messmikroskop), Abbe Komparator, Eppensteinprinzip; optische inkrementelle Encoder: Längenmessungen mit inkrementellen Encodern, Teilungsbreite vs. Detektorgröße, Moiré-Effekt, Prinzip eines optischen inkrementellen Encoders, Ermittlung Bewegungsrichtung Inkremental-Encoder, Quadratursignale und richtungsabhängige Zählung (Abtastplatte), Netzwerkinterpolatoren (Auflösungserhöhung), Demodulation für Encodersignale, Demodulationsabweichungen (Quantisierungs-, Amplituden-, Offset- und Phasenabweichungen), Heydemannkorrektur, Differenzsignale, Abtastung (abbildendes Prinzip, Durchlicht und reflektiertes Licht), kodierte Referenzmarken, Einfeldlesekopf, Abtastung (interferentielles Prinzip, reflektiertes Licht), Drei-Achsen-Verschiebungssensoren; optische absolute Encoder: absolut codierte Maßstäbe, V- und U-Anordnung und Gray Code, Pseudo Random Code; magnetische, induktive und kapazitive Linearencoder: magnetische Linearencoder, induktive Linearencoder, kapazitive Linearencoder; Längenmessgeräte: Universallängenmessgerät, Höhenmessgerät
- Längenmesstechnik (Interferometer): Interferenz und Interferometer: Interferometrie, Michelson Versuch, Interferenz, Wellengleichung, transversale elektromagnetische Welle (TEM), Polarisation des Lichtes, Überlagerung von Wellen (konstruktive und destruktive Interferenz), Voraussetzung für die interferometrische Längenmessung, Interferenz von Lichtwellen, Homodynprinzip, Heterodynprinzip, Interferenz am Michelson-Interferometer, Interferenz am Homodyninterferometer, Abstand der Interferenzlinien, Einteilung von Interferometern; Demodulation von Interferometersignalen: Demodulation am Homodyninterferometer, Demodulation am Heterodyninterferometer, Vergleich der Homodyn- und Heterodyninterferometer, Luftbrechzahl, parametrische und interferometrische Erfassung, Totstreckenkorrektur, praktische Realisierung der Demodulation am Homodyninterferometer, Quantisierungsabweichungen, Demodulationsabweichungen durch Quadratursignalrauschen, Längenabweichungen durch Offset-, Amplituden- und Phasenabweichungen, Kompensation der statischen Abweichungen, verbleibende dynamische Abweichungen; Kohärenz: räumliche und zeitliche Kohärenz, Kohärenzlänge von Einfrequenz- und Zweifrequenzlasern sowie Weißlicht; He-Ne-Laser und Rückführbarkeit: spontane und stimulierte Emission, Laser (Aufbau, Resonator und Entstehung der Lasermoden),

Resonatoranordnungen, Gauß-Strahlen, Transformation von Gauß-Strahlen (dünne Linsen), He-Ne-Laser (Energiezustände, Aufbau, Prinzip, Verstärkungskurve und Lasermoden, Frequenzstabilität), Methoden zur Stabilisierung von He-Ne-Lasern (Lamb-dip, externe Absorptionszelle, Intensitätsgleichheit bei Zeeman-Aufspaltung, Intensitätsgleichheit orthogonal linear polarisierter Moden), Messung der Beatfrequenz, optischer Frequenzkamm, Rückführbarkeit der Längenmessung (kurze Strecken), Realisierung der Meterdefinition, Rückführbarkeit der Längenmessung (große Strecken); Absolutinterferometrie: Mehrwellenlängeninterferometer; Interferometeraufbauten: Oberflächenspiegel, Prismen, Retroreflektoren, Strahlteiler, planparallele Platte, Drehkeilpaar, Linearpolarisatoren - strahlteilende Polarisatoren, Lambda/2- und Lambda/4-Platten, Faraday-Isolator, Baukastensysteme, Aufbauvarianten, Messabweichungen und Messkreise, Kompaktinterferometer (z. B. Homodyninterferometer), Kombination von Kippinvarianz und lateraler Verschiebung, Justage von Interferometern; Anwendung von Interferometern: Präzisions-Längenkomparator, Kalibrierinterferometer, Laser Tracer, Multilateration, Laser Vibrometrie, Interferenzkomparator

- Winkel- und Neigungsmesstechnik: Winkelmessung und Aufgaben: ebener Winkel, Raumwinkel, Messaufgaben; Winkelmaßverkörperungen: Einzelwinkelnormale, Winkelendmaße, Sinuslineal, Sinus-Winkel-Einstellgerät, Tangenslineal, Winkelprisma verstellbar, mechanische Kreisteilungsnormale, optische Kreisteilungsnormale, Winkelencoder (optisch oder induktiv), Spiegelpolygon, Pentaprisma; Winkelmessgeräte: Winkelmesser, Universalwinkelmesser, Winkelencoder (inkrementell absolut codiert); Messabweichungen: Scheitel- und Schenkeldeckung, Doppelablesung (180°-Ablesung); Neigungsmessung: Wasserwaagen, Libellen, Koinzidenzlibelle, Schlauchwaage, Klinometer/ Inklinometer (MEMs, Kraftkompensationssensoren); optische Winkelmessgeräte: Fernrohr, Kollimator, Strichplatten, Kollimator und Fernrohr, Autokollimator (visuelle und elektronische Ablesung), Autokollimator-Anwendungen (Winkelverschiebung, Geradheitsmessung, Rechtwinkligkeitsmessung, Kalibrierung von Drehtischen), Sextant, Theodolit und Tachymeter, Lasertracker, Winkelmessung mit Laserinterferometern, Kalibrierinterferometer
- Geometrische Produktspezifikation und Verifikation (GPS): Grundlagen der GPS: Systematik der Gestaltabweichungsarten (Maß-, Form-, Lageabweichungen und Abweichung der Oberflächenbeschaffenheit), Ordnungssystem für Gestaltabweichungen, geometrischen Toleranzen, Entwicklung der Normung und Messtechnik,

System der geometrischen Produktspezifikation, ISO-GPS-Matrix, Grundsätze, Dualitätsprinzip, Operatoren, Begriffsdefinition von Geometrieelementen (Nenn-, wirkliches, erfasstes und zugeordnetes Geometrieelement, …), Standardgeometrieelemente; Toleranzen von Längenmaßen: Größenmaße, Spezifikationsmodifizierer für Längenmaße, Toleranzen von Längenmaßen, Nennmaß, Grenzmaß, Abmaß, Grenzabmaß, ISO-Toleranzsystem für Längenmaße ISO-Passungen; Toleranzen von Winkelmaßen: Spezifikationsmodifizierer für Winkelmaße, Winkelgrößenmaße; Entscheidungsregeln für Konformitäts- und Nichtkonformitätsnachweis: Kennwerte für Messabweichungen, „Goldene Regel“ der Messtechnik nach Berndt (ca. 1924), Prüfung auf Konformität, Prüfung auf Nichtkonformität; Bezüge, Form-, Richtungs-, Orts- und Lauftoleranz, zusätzliche Spezifikationen (grundlegende GPS-Spezifikationen, Unabhängigkeitsprinzip, Maximum-Material-Bedingung, Minimum-Material-Bedingung, Reziprozitätsbedingung, Hüllbedingung, "Taylor'scher Grundsatz“, freier Zustand; Allgemeintoleranzen, Welligkeit und Rauheit, Kanten mit unbestimmter Gestalt, definierte Übergänge zwischen Geometrieelementen (Kante bestimmter Gestalt), Produktionsprozessspezifische Normen (Gußteile, Kunststoff-Formteile, thermisches Schneiden)

- Taktile Koordinatenmesstechnik: Historie, Gerätetechnik: Grundanordnung, konventionelle und unkonventionelle Bauarten, Gerätetechnik (Antriebe, Führungen, Längenmesssysteme), Tastsysteme (Übersicht, Messung der Auslenkung, Messsignale, Antastung, Einzelpunktantastung, Scanning, Richtungsempfindlichkeit, Erzeugung der Antastkraft, Kinematik, Bestandteile, kinematische Kopplungen, Dreh-Schwenk-System, Taster, Arten von Tastsystemen, mechanische Filterwirkung), Steuereinheit, Zusatzeinrichtungen (Drehtisch, Taster- und Messkopfwechselbank, Werkstückfixierung); Vorbereitung, Durchführung und Auswertung der Messung: Beschreiben und Festlegen der Messaufgabe inkl. Bezugssystem, Feststellen Einflüsse auf das Messergebnis, Vorbereitung der Messung, Aufspannen des Werkstücks, Auswahl des Messkopfes und Tasters, Einmessen des Tasters, Festlegen der Messstrategie, Auswertung der Messergebnisse (Ausgleichsverfahren, Operatoren, Messunsicherheitsbestimmung); Spezifikation, Parameter und Prüfung (Annahme- und Bestätigungsprüfung, Überwachung von Koordinatenmessgeräten, Normale, Spezifikation)
- Taktile Oberflächenmesstechnik: Oberflächen, Charakterisierung von Oberflächen, Oberflächenmessprinzipien, Wechselwirkung und Einflussgrößen, Oberflächenmessverfahren; taktile

Messverfahren: Tastschnittgeräte, Diamant-Tastspitze, Messumformer, morphologische Filterwirkung, Bauarten; Überblick Oberflächenparameter; Profilparameter (2D; DIN EN ISO 4287 und DIN EN ISO 21920-2): Auswertung eines Oberflächenprofils, Filterung, Messstrecke und Einzelmessstrecken, Senkrechtkenngrößen, Waagerechtkenngrößen, gemischte Kenngrößen, Kenngrößen aus charakteristischen Kurven, Motifkenngrößen; Flächenparameter (3D; DIN EN ISO 25178-2): Auswertung einer Flächentopographie, Höhenparameter, Hybridparameter, flächenhafte Materialanteilkurve, Topographische Elemente; Streulichtparameter: Varianz der Verteilungskurve

Content:

- Basics: Sub-areas of industrial metrology, basic tasks of manufacturing metrology, measuring conditions and points in time, methods and subtasks of manufacturing metrology, objectives of manufacturing metrology; definitions of terms: measuring, monitoring, testing, checking, gauging, history of manufacturing metrology, equipment in manufacturing metrology, basic classification of measuring and testing equipment, classical manufacturing metrology, coordinate metrology; terms of metrology (repetition from fundamental lecture): measured quantity, quantity value, measurement result, measured value, measurement principle, measurement method, measurement procedure, sensitivity, measurement range, resolution (spatial or scale resolution vs. structural resolution, amplitude-wavelength diagram), true value, agreed value, systematic and random measurement deviation, calibration, verification, validation, measurement precision, measurement accuracy, measurement correctness, measurement uncertainty
- Length measuring technique (hand-held measuring devices and standards): tasks and use of length measuring technique, caliper (construction, reading), vernier, parallax deviation, error of the 1st order, Abbe's comparator principle, measuring variants with calipers, types of calipers, micrometers (construction, reading), error of the 2nd order, types of micrometers, dial gauge, vernier pointer, lever gauge, inductive probes (construction, characteristic curve), causes of measuring errors: measuring circuit, temperature influences, expansion compensation, surface contact pressure and flattening, deformation of measuring plates and long parts, tilting and guiding deviations, shape deviations and changes (equal thickness or Reuleaux polygons), ellipse and three-arc equal thickness, three-point measurement, centring errors and centring aids; materials for measuring circuits: Aluminium, steel, Invar 36, Super Invar 32-5, natural stone, polymer concrete, ceramics, sintered silicon carbide, NEXCERA N113G, titanium silicate glass ULE, Zerodur, mechanical

		stresses and creep; Dimensional standards: gauge blocks, feeler gauges, limit gauges • Length measuring technique (scales and encoders): scales with visual reading: scales with graduations, resolving power of the eye, spiral eyepiece, 1st and 2nd order error (measuring microscope), Abbe comparator, Eppenstein principle; optical incremental encoders: length measurement with incremental encoders, graduation width vs. detector size, Moiré effect, principle of an optical incremental encoder, determination of direction of movement incremental encoder, quadrature signals and direction-dependent counting (scanning plate), network interpolators (resolution increase), demodulation for encoder signals, demodulation deviations (quantisation, amplitude, offset and phase deviations), Heydemann correction, differential signals, scanning (imaging principle, transmitted and reflected light), coded reference marks, single-field reading head, scanning (interferential principle, reflected light), three-axis displacement sensors; optical absolute encoders: absolute coded scales, V and U arrangement and Gray code, pseudo random code; magnetic, inductive and capacitive linear encoders: magnetic linear encoders, inductive linear encoders, capacitive linear encoders; linear encoders: universal linear encoder, height encoder • Length measurement technique (interferometer): interference and interferometer: interferometry, Michelson experiment, interference, wave equation, transverse electromagnetic wave (TEM), polarisation of light, superposition of waves (constructive and destructive interference), prerequisite for interferometric length measurement, interference of light waves, homodyne principle, heterodyne principle, interference at the Michelson interferometer, interference at the homodyne interferometer, distance of interference lines, classification of interferometers; demodulation of interferometer signals: demodulation at the homodyne interferometer, demodulation at the heterodyne interferometer, comparison of homodyne and heterodyne interferometers, air refractive index, parametric and interferometric acquisition, dead-pahth correction, practical realisation of demodulation at the homodyne interferometer, quantisation deviations, demodulation deviations due to quadrature signal noise, length deviations due to offset, amplitude and phase deviations, compensation of static deviations, remaining dynamic deviations; coherence: spatial and temporal coherence, coherence length of single-frequency and dual-frequency lasers and white light; He-Ne laser and traceability: spontaneous and stimulated emission, lasers (structure, resonator and origin of laser modes), resonator arrangements, Gaussian beams, transformation of Gaussian beams (thin lenses), He-Ne lasers (energy states, structure, principle, gain curve and laser modes,

frequency stability), methods for stabilising He-Ne lasers (Lamb-dip, external absorption cell, intensity equality with Zeeman splitting, intensity equality of orthogonally linearly polarised modes), measurement of beat frequency, optical frequency comb, traceability of length measurement (short distances), realisation of metre definition, traceability of length measurement (long distances); absolute interferometry: multi-wavelength interferometer; interferometer set-ups: surface mirrors, prisms, retroreflectors, beam splitters, plane-parallel plate, rotating wedge pair, linear polarisers - beam-splitting polarisers, lambda/2 and lambda/4 plates, Faraday isolator, modular systems, set-up variants, measurement errors and measurement circuits, compact interferometers (e.g. homodyne interferometer), combination of tilt invariance and lateral displacement, adjustment of interferometers; application of interferometers: precision length comparator, calibration interferometer, laser tracer, multilateration, laser vibrometry, interference comparator

- Angle and inclination measuring technology: angle measurement and tasks: plane angle, solid angle, measuring tasks; angle measuring standards: single angle standards, angle end measures, sine ruler, sine angle adjuster, tangent ruler, angle prism adjustable, mechanical circular graduation standards, optical circular graduation standards, angle encoder (optical or inductive), mirror polygon, pentaprism; angle measuring instruments: protractor, universal protractor, angle encoder (incremental absolute coded); measurement deviations: vertex and limb coverage, double reading (180° reading); inclination measurement: spirit levels, bubble levels, coincidence bubble, hose level, clinometer/ inclinometer (MEMs, force compensation sensors); optical angle measuring instruments: Telescope, collimator, graticules, collimator and telescope, autocollimator (visual and electronic reading), autocollimator applications (angular displacement, straightness measurement, squareness measurement, calibration of rotary tables), sextant, theodolite and tachymeter, laser tracker, angle measurement with laser interferometers, calibration interferometer
- Geometric product specification and verification (GPS): fundamentals of GPS: systematics of shape deviation types (dimensional, form, positional and surface quality deviations), classification system for shape deviations, geometric tolerances, development of standardisation and metrology, system of geometric product specification, ISO GPS matrix, principles, duality principle, operators, definition of terms of geometry elements (nominal, real, recorded and assigned geometry element, ...), standard geometry elements; tolerances of length dimensions: size dimensions, specification modifiers for length dimensions, tolerances of length

dimensions, nominal dimension, limit dimension, allowance, limit allowance, ISO tolerance system for length dimensions ISO fits; tolerances of angle dimensions: specification modifiers for angular dimensions, angular size dimensions; decision rules for proof of conformity and non-conformity: characteristic values for measurement deviations, "Golden Rule" of metrology according to Berndt (ca. 1924), verification of conformity, verification of non-conformity; references, shape, direction, location and running tolerance, additional specifications (basic GPS specifications, independence principle, maximum material condition, minimum material condition, reciprocity condition, envelope condition, "Taylor's principle", free state; general tolerances, waviness and roughness, edges of indeterminate shape, defined transitions between geometry elements (edge of determinate shape), production process specific standards (castings, moulded plastic parts, thermal cutting)

- Tactile coordinate measuring technology: history, instrument technology: basic arrangement, conventional and unconventional designs, machine technology (drives, guideways, length measuring systems), tactile systems (overview, measurement of deflection, measuring signals, probing, single-point probing, scanning, directional sensitivity, generation of probing force, kinematics, components, kinematic couplings, rotary-tilt system, probes, types of tactile systems, mechanical filter effect), control unit, additional equipment (rotary table, probe and measuring head changing bench, workpiece fixing); preparation, execution and evaluation of the measurement: describing and specifying the measuring task incl. reference system reference system, determining influences on the measurement result, preparing the measurement, clamping the workpiece, selecting the measuring head and probe, calibrating the probe, determining the measurement strategy, evaluating the measurement results (compensation methods, operators, determining the measurement uncertainty); specification, parameters and testing (acceptance and confirmation testing, monitoring coordinate measuring machines, standards, specification)
- Tactile surface metrology: surfaces, characterisation of surfaces, surface measuring principles, interaction and influencing variables, surface measuring methods; tactile measuring methods: tactile measuring methods: stylus instruments, diamond stylus tip, transducer, morphological filter effect, types; overview of surface parameters; profile parameters (2D; DIN EN ISO 4287 and DIN EN ISO 21920-2): evaluation of a surface profile, filtering, measuring section and individual measuring sections, perpendicular parameters, horizontal parameters, mixed parameters, parameters from characteristic curves, motif parameters; surface parameters

		(3D; DIN EN ISO 25178-2): evaluation of an area topography, height parameters, hybrid parameters, area material proportion curve, topographic elements; scattered light parameters: variance of the distribution curve
6	**Lernziele und Kompetenzen**	Wissen • Die Studierdenden können die Motivation, Ziele, Grundsätze und Strategien der Fertigungsmesstechnik darlegen. • Die Studierenden können die operative Herangehensweise an Aufgaben der messtechnischen Erfassung von dimensionellen und geometrischen Größen an Werkstücken nennen. • Die Studierdenden können Messaufgaben, deren Durchführung und Auswertung von Messungen beschreiben. Verstehen • Die Studierenden können Messergebnisse und das zugrunde liegenden Verfahren angemessen kommunizieren und interpretieren. Anwenden • Die Studierenden können Messaufgaben durch das Erlernte implementieren. • Die Studierenden können geeignete Verfahren im Bereich Fertigungsmesstechnik eigenständig auswählen. Analysieren • Die Studierenden können Messaufgaben in der Fertigungsmesstechnik beurteilen und analysieren. • Die Studierenden können Schwachstellen in der Planung und Durchführung selbstständiges erkennen. • Die Studierenden können Messergebnissen aus dem Bereich der Fertigungsmesstechnik bewerten
7	**Voraussetzungen für die Teilnahme**	Für eine optimale Vorbereitung empfiehlt sich eine Belegung des Moduls "Grundlagen der Messtechnik". Dies ist jedoch keine Teilnahmevorraussetzung für das Modul "Fertigungsmesstechnik I".
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch

16	**Literaturhinweise**	• Kohlrausch, Friedrich: Praktische Physik : zum Gebrauch für Unterricht, Forschung und Technik. Band 1-3, 24. Auflage, Teubner Verlag, 1996 ISBN 3-519-23001-1, 3-519-23002-X, 3-519-23000-3 • DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Beuth Verlag GmbH, 3. Auflage 2010 • Pfeifer, Tilo: Fertigungsmeßtechnik. R. Oldenbourg Verlag München Wien, 1998 ISBN 3-486-24219-9 • Keferstein, Claus P.: Fertigungsmesstechnik. 7. Auflage, Vieweg+Teubner Verlag, 2011 ISBN 978-3-8348-0692-5 • Warnecke, H.-J.; Dutschke, W.: Fertigungsmeßtechnik. Springer-Verlag Berlin Heidelberg New York Tokyo, 1984 ISBN 3-540-11784-9 • Christoph, Ralf; Neumann, Hans Joachim: Multisensor-Koordinatenmesstechnik. 3. Auflage, Verlag Moderne Industrie, 2006 ISBN 978-3-937889-51-2 • Neumann, Hans Joachim: Koordinatenmesstechnik im der industriellen Einsatz. Verlag Moderne Industrie, 2000 ISBN 3-478-93212-2 • Ernst, Alfons: Digitale Längen- und Winkelmesstechnik. 4. Auflage, Verlag Moderne Industrie, 2001 ISBN 3-478-93264-5 • Joza, Jan: Messen großer Längen. VEB Verlag Technik Berlin, 1969 • Henzold, Georg: Form und Lage. 3. Auflage, Beuth Verlag GmbH Berlin, 2011 ISBN 978-3-410-21196-9 • Weckenmann, A.: Koordinatenmesstechnik: Flexible Strategien für funktions- und fertigungsgerechtes Prüfen, 2. Auflage, Carl Hanser Verlag München, 2012 • *Internetlinks für weitere Information zum Thema Messtechnik* • [Video des VDI: Messtechnik - Unsichtbare Präszision überall]http://youtu.be/tQgvr_Y3Gl0 • [Multisensor-Koordinatenmesstechnik]http://www.koordinatenmesstechnik.de/ • [E-Learning Kurs AUKOM Stufe 1]http://www.aukom-ev.de/deutsch/elearning/content.html

1	**Modulbezeichnung** 96925	**Fertigungsmesstechnik II** Manufacturing metrology II	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Tino Hausotte
5	**Inhalt**	*Optische Oberflächenmesstechnik:* Überblick Oberflächenabweichungen und Oberflächenmessprinzipien, Wechselwirkungen, Einteilung der optischen Oberflächenmessverfahren, Mikroskope und Komponenten, Messmikroskope, Numerische Apertur, Punktverwaschungsfunktion, Auflösungsvermögen, Modulations-Transfer-Funktion Fokusvariation, Fokusvariation mit strukturierter Beleuchtung, Flying Spot Mikroskop, konfokales Mikroskop (Aufbau, Prinzip, Kennlinie, Nipkow-Scheibe, Scanspiegel, Mikrolinsenarray), Laserscanningmikroskop, konfokaler zwei Wellenlängenfasersensor, chromatischer Weißlichtsensor, Laser-Autofokusverfahren, Interferenzmikroskope (Michelson, Mirau, Linnik, Phasenschieber), Weißlichtinterferometer Streulichtmessung *Taktile Formmesstechnik:* Grundlagen der Formmesstechnik, Prinzip, Charakteristika, Messaufgaben Bauarten von taktilen Formmessgeräten (Drehtisch-, Drehspindelgeräte, Universalmessgeräte, Tastsysteme) Messabweichungen (Einflussfaktoren, Abweichungen der Drehführung und deren Bestimmung, Abweichungen der Geradführungen) Kalibrierung von Formessgeräten (Flick-Normale, Vergrößerungsnormale, Kugelnormale, Mehrwellennormale) Mehrlagenverfahren, Umschlagverfahren *Optische Formmesstechnik:* Interferometrische Formmessung (Interferenz gleicher Neigung und gleicher Dicke, Mehrstrahlinterferenz, Fabry-Perot und Fizeauinterferometer, Interferenzfilter, Newtonsche Ringe, Phasenschiebeinterferometer, Demodulation mit Phasenschiebung, synthetische Wellenlänge, Anwendung der Fizeau-Interferometrie, Einfluss der Referenzfläche, Dreiplattentest, Interferometrie streifendem Einfall, Twyman-Green Interferometer, Einsatzgrenzen) Deflektometrische Formmessung (Überblick Deflektometrie, Grundprinzip, Extended Shear Angle Difference Methode, flächenhafte Deflektometrie, Einsatzgrenzen) *Photogrammmetrie:* Grundprinzip, Stereophotogrammmetrie, passive Triangulation, Grundlagen, aktive Triangulation (Punkttriangulation, linienhafte und flächenhafte Triangulation) Streifenlichtprojektion (strukturierte Beleuchtung, Grundprinzip Ein- und Zweikamerasysteme, Kodierung Gray Code, Phasenschiebung, Kombinierte Beleuchtung aus Gray Code und Phasenschiebung, Anwendung, Datenverarbeitung, Einsatzgrenzen) *Röntgen-Computertomografie:* Röntgenstrahlung, Grundprinzip der Röntgen-Computertomografie, Aufbau und Scanvarianten, Röntgenstrahlquellen, Strahlungsspektrum, Wechselwirkung mit Material (Photoelektrischer Effekt, Compton Streuung), Detektoren, Vergrößerung, Rekonstruktion (Radontransformation,

		algebraische Rekonstruktion, gefilterte Rückprojektion, Artefakte (Strahlaufhärtung, Ringartefakte, Streustrahlung, Scannerausrichtung), Schwellwertfindung, Anwendung (Defekterkennung, Micro- und Nano-CT, Hochenergie-CT, Multimaterial), Rückführung *Spezifikation und Messung optischer Komponenten:* Zeichnungen für optische Elemente und Systeme, Materialspezifikation, Spezifikation von Oberflächenformtoleranzen, Prüfung der Oberflächenformabweichungen (Passe) mit Probegläsern, Oberflächenbehandlungen und Beschichtungen, Messung geometrischer Spezifikationen *Mikro- und Nanomesstechnik:* Positioniersysteme (Führungen und Antriebe, Gewichtskraftkompensation), metrologischer Rahmen und Gerätekoodinatensysteme, Antastprinzipien und Messsystem (Rasterelektronenmikroskop, Rastertunnelmikroskop, Rasterkraftmikroskope, Nahfeldmikroskope, mikrotaktile Antastung), Mikro- und Nanokoordinatenmesssysteme, Einflussgrößen, Kalibrierung und Rückführung
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen • Die Sudierenden kennen relevante Definitionen, Fachbegriffe und Kriterien der Fertigungsmesstechnik. • Die Studierenden können einen Überblick zur Gerätetechnik der Fertigungsmesstechnik sowie deren Funktionsweise und Einsatzgebiete wiedergeben • Die Studierenden wissen um die operative Herangehensweise an Aufgaben der messtechnischen Erfassung von dimensionellen und geometrischen. Verstehen • Die Studierenden sind in der Lage die, den vorgestellten Messgeräten der Fertigungsmesstechnik, zugrundeliegenden Messprinzipien in eigenen Worten zu erläutern. • Die Studierenden können Messaufgaben beschreiben und interpretieren, und Schwachstellen in der Planung und Durchführung erkennen. • Die Studierenden können Messergebnisse und die zugrunde liegenden Verfahren angemessene kommunizieren und interpretieren. Anwenden • Die Studierenden können eigenständig geeignete Verfahren im Bereich Fertigungsmesstechnik auswählen. • Die Studierenden können das Erlernte auf unbekannte, aber ähnliche Messaufgaben transferieren. Evaluieren (Beurteilen) • Die Studierenden können Messaufgaben in der Fertigungsmesstechnik beurteilen und strukturell analysieren. • Die Studierenden sind in der Lage Messergebnisse zu hinterfragen und auf dieser Basis die Funktionalität des Messsystems sowie die zum Zeitpunkt der Messung vorherrschenden Messbedingungen zu bewerten. Erschaffen

		• Die Studierenden können die Eignungsuntersuchungen verschiedener Messprinzipien zur Erfüllung neuer Messaufgaben erstellen und auf deren Basis adaptierte Messsysteme konzipieren.
7	**Voraussetzungen für die Teilnahme**	Eine Teilnahme an der Lehrveranstaltung "Fertigungsmesstechnik 1" wird empfohlen, ist jedoch keine Teilnahmevorraussetzung.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	International Vocabulary of Metrology Basic and General Concepts and Associated Terms, VIM, 3rd edition, JCGM 200:2008, http://www.bipm.org/en/publications/guides/vim.html DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Korrigierte Fassung 2012, Beuth Verlag GmbH, 4. Auflage 2012 Pfeifer, Tilo: Fertigungsmeßtechnik. R. Oldenbourg Verlag München Wien, 1998 ISBN 3-486-24219-9 Keferstein, Claus P.: Fertigungsmesstechnik. 7. Auflage, Vieweg +Teubner Verlag, 2011 ISBN 978-3-8348-0692-5 Warnecke, H.-J.; Dutschke, W.: Fertigungsmeßtechnik. Springer-Verlag Berlin Heidelberg New York Tokyo, 1984 ISBN 3-540-11784-9 Christoph, Ralf; Neumann, Hans Joachim: Multisensor-Koordinatenmesstechnik. 3. Auflage, Verlag Moderne Industrie, 2006 ISBN 978-3-937889-51-2 Neumann, Hans Joachim: Koordinatenmesstechnik im der industriellen Einsatz. Verlag Moderne Industrie, 2000 ISBN 3-478-93212-2

		Weckenmann, A.: Koordinatenmesstechnik: Flexible Strategien für funktions- und fertigungsgerechtes Prüfen, 2. Auflage, Carl Hanser Verlag München, 2012 Hausotte, Tino: Nanopositionier- und Nanomessmaschinen - Geräte für hochpräzise makro- bis nanoskalige Oberflächen- und Koordinatenmessungen. Pro Business Verlag, 2011 - ISBN 978-3-86805-948-9 David J. Whitehouse: Handbook of Surface and Nanometrology, Crc Pr Inc., 2010 - ISBN 978-1420082012

1	**Modulbezeichnung** 607629	**Hauptseminar Messtechnik** Advanced seminar Manufacturing metrology	**2,5 ECTS**
2	Lehrveranstaltungen	Seminar: Hauptseminar Fertigungsmesstechnik (2 SWS)	-
3	Lehrende	Prof. Dr.-Ing. Tino Hausotte	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Tino Hausotte

5	**Inhalt**	*Ablauf des Seminars*
		|*1. Voranmeldung StudOn*|
		•
		  ° Die Anmeldung zum Hauptseminar erfolgt in der Regel am Anfgang des Semsters. Ausnahmen sind möglich.
		  ° Hierfür wird eine Liste der Seminarthemen mit zugeordnete StudOn-Gruppen bereit gestellt.
		  ° Die Anmeldung zu einem bestimmten Thema erfolgt durch selbstständige Anmeldung zur zugeordneten StudOn-Gruppe.
		  ° Kontakt mit dem Betreuuer innerhalb der ersten Woche nach anmeldung notwendig.
		  ° Klärung von Ziel, Auftrag und Kontext.
		  ° Recherche, Auswahl der Informationen.
		  ° Grobe Ablaufplanung der Präsentation (Begrüßung und Themenübersicht, Einstieg ins Thema, Transport der Inhalte, Themenbegrenzung), Ausstieg, Fragen und Diskussion).
		  ° Feine Ablaufplanung: Detaillierung der Inhalte (Sinnvolle Gliederung, Inhaltlichen Fortgang visualisieren, Zum Thema immer wieder zurückkehren, Gedankensprünge vermeiden, Foliensprünge vermeiden, Layout für den roten Faden", Ringschluss zwischen Anfang und Ende schaffen).
		  ° Erstellten der Präsentation (Vorlage auf StudOn beachten).
		  ° Terminplan der Präsentationen wird vom Koordinator festgelegt und per E-Mail mitgeteilt (Termine sind in der Regel gegen Ende der Vorlesungszeit). Ausnahmen sind möglich.
		  ° Termin zur Abgabe der Präsentation: eine Woche vor dem Präsentationstermin.
		  ° Durchführung der Präsentation (Präsentationsdauer 20 min. + 10 min. Diskussion)
		  ° Teilnahme an 5 weiteren Vorträgen.
		  ° Notenbekanntgabe direkt nach der Präsentation.
		  ° Koordinator schickt den ausgestellten Schein direkt an das Prüfungsamt.
		  ° Auf Anfrage Feedback vom Betreuer (sofern gewünscht).

6	**Lernziele und Kompetenzen**	Die Studierenden: • erlangen grundlegender Kenntnisse in Recherche, Themenaufbereitung und Präsentationstechniken,

		• erarbeiten Schwerpunkte technischer Zusammenhänge bei einem gegebenen Thema, • vertiefen eigenständig einen technischen Schwerpunkt an Hand eines konkreten Beispiels der Fertigungsmesstechnik, • erlernen die Fähigkeit, sich in unbekannte Probleme einzuarbeiten und diese verständlich zu präsentieren, • erlernen die Fähigkeit, als Zuhörer aktiv Fragen zu formulieren und technische Sachverhalte zu diskutieren,
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Variabel Die Gesamtmodulnote errechnet sich aus der Vortragsleistung. Im Zuge des Hauptseminars ist ein Thema auszuarbeiten und in einem 20 minütigem Vortrag zu präsentieren. Für das Bestehen des Moduls sind zusätzlich 5 Vorträge anzuhören. Die möglichen Themen werden auf StudOn bereitgestellt. Die Vortragsdauer beträgt 20 Minuten mit anschließender 10 minütiger Diskussion.
11	**Berechnung der Modulnote**	Variabel (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 45 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 97248	**Prozess- und Temperaturmesstechnik** Process and temperature metrology	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Prozess- und Temperaturmesstechnik (2 SWS) Übung: Prozess- und Temperaturmesstechnik - Übung (2 SWS)	- -
3	Lehrende	Prof. Dr.-Ing. Tino Hausotte	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Tino Hausotte
5	**Inhalt**	• Temperaturmesstechnik: Messgröße Temperatur: (thermodynamische Temperatur, Symbole, Einheiten, Neudefinition der SI Einheiten, Temperatur als intensive Größe, Prinzip eines Messgerätes, direkte Messung und Voraussetzungen, indirekte Temperaturmessung und Voraussetzungen, Überblick primäre Temperaturmessverfahren, unmittelbar und mittelbare Temperaturmessung) Prinzipielle Einteilung der Temperaturmessverfahren, Temperaturskalen: praktische Temperaturskalen (Tripelpunkte, Schmelz- und Erstarrungspunkte), klassische Temperaturskalen (Benennung und Fixpunkte), ITS 90 (Bereich, Fixpunkte, Interpolationsinstrumente) Grundlagen der Temperaturmessung mit Berührungsthermometer Mechanische Berührungsthermometer Widerstandsthermometer (Pt100, NTC, PTC, Kennlinie, Messschaltungen) Thermoelemente (Grundlagen, Aufbau, Vergleichsstelle, Bauformen) Spezielle Temperaturmessverfahren (Rauschtemperaturmessung, Quarz-Thermometer) Strahlungsthermometer (Grundlagen, Prinzip, Schwarzer Strahler) • Wägetechnik: Messgrößen Masse und Gewicht, Prototypen, Rückführung und Masseableitung, Neudefinition des kg, Einflüsse auf Massenmessung, Balkenwaagen, Federwaagen, Elektromagnetische Kraftkompensationswaage, Komparatoren • Messen der Dichte: Messgröße Dichte, Einteilung der Dichtemessverfahren, Messverfahren für feste, flüssige und gasförmige Stoffe • Messen des Druckes: Messgröße Druck, Einteilung der Druckmessverfahren, Druckwaagen, Flüssigkeitsmanometer und Barometer, federelastische Druckmessgeräte, Druckmessumformer, Druckmittler, piezoelektrische Druckmessgeräte • Messen des Durchflusses: Messgröße Durchfluss, Einteilung der Durchflussmessverfahren, Volumetrische Messverfahren, Massendurchflussmessung • Messen des Füllstandes und Grenzstandes: Grundlagen (Messgrößen Füllstand und Grenzstand, Behälter, Einteilung), Messverfahren

		• Messen der Feuchte: Grundlagen (Messgröße Feuchte), Gasfeuchtemessung, Materialfeuchtemessung **<u>Content:</u>** • Temperature measurement: Measure "temperature (thermodynamic temperature, symbols, units, temperature and intensive quantity, principle of a measuring instrument, and direct measurement conditions, indirect temperature measurement and conditions Overview primary temperature measurement methods, direct and indirect temperature measurement) Basic classification of temperature measurement methods Temperature scales: practical temperature scales (triple points, melting and solidification points), classical temperature scales (naming and fixed points), ITS 90 (range, fixed points, interpolating instruments) Mechanical contact thermometers Resistance thermometer (Pt100, NTC, PTC, characteristic, measurement circuits) Thermocouples (foundations, structure, junction, mounting positions) Special methods of temperature measurement (noise temperature measurement, quartz thermometer) Pyrometer Static and dynamic thermal sensors • Weighing technology: Mass and weight, prototypes, traceability of mass, new definition of the kg, influences on mass measurement, beam balances, spring scales, electromagnetic force compensation, comparators • Measurement of density: Measurand density, Classification of density measurement methods, measurement procedures for solid, liquid and gaseous substances • Measurement of pressure: Measurand pressure, Classification of pressure measuring method, Pressure balances Liquid manometers and barometers, Resilient pressure gauges, Pressure transmitters, Diaphragm seals, Piezoelectric pressure gauge • Measurement of flow: Measurand flow, Classification of flow measurement methods, Volumetric measurement methods, Mass flow measurement • Measurement of filling level and limit state: Fundamentals (Measurands filling level and limit state, tanks, classification), Measuring methods • Measurement of humidity: Fundamentals (Measurand humidity), Gas humidity measurement, Material humidity measurements
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen • Die Studierenden kennen die Motivation, Ziele, Grundsätze und Strategien der Prozessmesstechnik. • Die Studierenden können Messaufgaben, die Durchführung und Auswertung von Messungen beschreiben. Verstehen

		• Die Studierenden können Messergebnissen und der zugrundeliegenden Verfahren angemessen kommunizieren und interpretieren. • Die Studierenden verstehen die operative Herangehensweise an Aufgaben der messtechnischen Erfassung von nicht-geometrischen Prozessgrößen. Anwenden • Die Studierenden können Messaufgaben in den genannten Bereichen analysieren und beurteilen. • Die Studierenden können Messergebnissen aus dem Bereich Prozessmesstechnik bewerten. • Die Studierenden können geeignete Verfahren im Bereich Prozess- und Temperaturmesstechnik eigenständig auswählen. Analysieren • ° Die Studierenden können das Erlernte auf unbekannte, aber ähnliche Messaufgaben übertragen.
7	**Voraussetzungen für die Teilnahme**	• Der Besuch der Grundlagen-Vorlesungen |Grundlagen der Messtechnik| (GMT) wird empfohlen.
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Hoffmann, Jörg: Handbuch der Messtechnik. 4. Auflage, Carl Hanser Verlag München, 2012 ISBN 978-3-446-42736-5 • Bernhard, Frank: Technische Temperaturmessung. Springer-Verlag Berlin Heidelberg New York, 2004 ISBN 3-540-62672-7 • Freudenberger, Adalbert: Prozeßmeßtechnik. Vogel Buchverlag, 2000 ISBN 978-3802317538 • Kohlrausch, Friedrich: Praktische Physik : zum Gebrauch für Unterricht, Forschung und Technik. Band 1-3, 24. Auflage, Teubner Verlag, 1996 ISBN 3-519-23001-1, 3-519-23002-X, 3-519-23000-3 • DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie Grundlegende und allgemeine Begriffe und zugeordnete

		Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Beuth Verlag GmbH, 3. Auflage 2010 *Internetlinks für weitere Information zum Thema Messtechnik* • [Video des VDI: Messtechnik - Unsichtbare Präszision überall]http://youtu.be/tQgvr_Y3Gl0

1	**Modulbezeichnung** 23030	**Qualitätsmanagement II - Phasenübergreifendes Qualitätsmanagement** Phasenübergreifendes Qualitätsmanagement (QM II)	**2,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Qualitätsmanagement QMaK (2 SWS)	-
3	Lehrende	Prof. Dr.-Ing. Tino Hausotte	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Tino Hausotte
5	**Inhalt**	• Qualitätsmanagementsystem - Auditierung und Zertifizierung • Total Quality Management und EFQM-Modell • Ausbildung und Motivation • Kontinuierliche Verbesserungsprogramme und Benchmarking • Problemlösungstechniken und Qualitätszirkel • Qualitätsbewertung • Qualität und Wirtschaftlichkeit • Six Sigma • Qualitätsmanagement bei Medizinprodukten • |Qualitätsbewertung (Übung)| • |Qualitätsbezogene Kosten und Wirtschaftlichkeit (Übung)|
6	**Lernziele und Kompetenzen**	Nach dem Besuch der Veranstaltung sind die Teilnehmenden in der Lage, Wissen: • o Wissen zu Qualitätsmanagement als unternehmens- und produktlebenszyklusübergreifende Strategie zu veranschaulichen o Anforderungen, Aufbau, Einführung sowie die Auditierung und Zertifizierung von Qualitätsmanagementsystemen darzustellen o Business Excellence anhand Total Quality Management (TQM), Unternehmensbewertungsmodelle wie EFQM und kontinuierlicher Verbesserungsprozesse im Unternehmen auszuführen o die Wirtschaftlichkeit von Qualitätsverbesserungsmaßnahmen zu demonstrieren o die Methodik Six Sigma" zu beschreiben und dem Kontext der Qualitätsverbesserung zuzuordnen o Handlungsgrundlagen hinsichtlich Ausbildungs-, Motivations- und Organisationsverbesserung zu ermitteln Evaluieren: die Qualität mit etablierten Vorgehensweisen zu bewerten
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik Bachelor of Science Informatik 20222

10	**Studien- und Prüfungsleistungen**	schriftlich (60 Minuten) Klausur mit MultipleChoice, Dauer (in Minuten): 60
11	**Berechnung der Modulnote**	schriftlich (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 45 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Kamiske, G. F.; Brauer, J.-P.: Qualitätsmanagement von A - Z, Carl Hanser Verlag, München 2011 • Pfeifer, T.; Schmitt, R.: Masing Handbuch Qualitätsmanagement, Hanser, München 2021 • Wagner, K. W.; Patzak, G.: Performance Excellence - Der Praxisleitfaden zum effektiven Prozessmanagement, Carl Hanser Verlag, München 2020 • Zink, K. J.: Mitarbeiterbeteiligung bei Verbesserungs- und Veränderungsprozessen, Carl Hanser Verlag, München 2007

1	**Modulbezeichnung** 95940	**Qualitätsmanagement I - Qualitätstechniken für die Produktentstehung** Quality management I - Quality engineering in the product development process	**2,5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Qualitätstechniken - QTeK - vhb (2 SWS)	-
3	Lehrende	Prof. Dr.-Ing. Tino Hausotte	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Tino Hausotte
5	**Inhalt**	• Einführung und Begriffe • Grundwerkzeuge des Qualitätsmanagements • Erweiterte Werkzeuge des Qualitätsmanagements • Qualitätsmanagement in der Produktplanung (QFD) • Qualitätsmanagement in der Entwicklung und Konstruktion (DR, FTA, ETA, FMEA) • Versuchsmethodik • Maschinen- und Prozessfähigkeit, Qualitätsregelkarten • Zuverlässigkeitstechniken • Qualitätsmanagementsystem - Aufbau und Einführung • |Grundwerkzeuge des QM (Einsendeaufgabe)| • |QFD und FMEA (Einsendeaufgabe)| • |Versuchsmethodik (Einsendeaufgabe)| • |SPC (Einsendeaufgabe)|

6	**Lernziele und Kompetenzen**	Nach dem Besuch der Veranstaltung sind die Teilnehmenden in der Lage,
		Wissen:
		•
		  ° die Werkzeuge, Techniken und Methoden des Qualitätsmanagements entlang des Produktlebenszyklus darzustellen
		  ° die Zuverlässigkeit von Systemen zu beschreiben
		  ° den Aufbau und die Einführung von Qualitätsmanagementsystemen darzustellen
		  ° die grundlegenden Qualitätsmethoden, -techniken und -werkzeuge auf ein anderes Problem zu übertragen
		  ° Prozesse mit Hilfe der statistischen Prozesslenkung (SPC), Qualitätsregelkarten und Prozessfähigkeitsindizes zu beschreiben
		  ° mit Hilfe der Qualitätsmethoden, -techniken und -werkzeugen Probleme zu analysieren
		  ° statistische Versuchspläne auf praktische Probleme zu übertragen und aus den Ergebnissen die Zusammenhänge und Einflüsse der Faktoren zu interpretieren
		  ° statistische Auswertungen zu interpretieren und neue Probleme auf statistische Auffälligkeiten zu testen

7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!

9	**Verwendbarkeit des Moduls**	Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten) Klausur mit MultipleChoice, Dauer (in Minuten): 60
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 30 h Eigenstudium: 45 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• ° DIN (Hrsg.): Internationales Wörterbuch der Metrologie, Beuth-Verlag, Berlin 1994 Masing, W.: Handbuch Qualitätsmanagement, Carl Hanser Verlag, München 2007

1	**Modulbezeichnung** 96930	**Rechnergestützte Messtechnik** Computer-aided metrology	**5 ECTS**
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Tino Hausotte
5	**Inhalt**	*Grundlagen:* Grundbegriffe (Größe, Größenwert, Messgröße, Maßeinheit, Messprinzip, Messung, Messkette, Messsignal, Informationsparameter, analoges und digitales Signal) Prinzip eines Messgerätes, direkte und indirekte Messmethode, Kennlinie und Kennlinienarten, analoge und digitale Messmethoden, kontinuierliche und diskontinuierliche Messung, Zeit- und Wertdiskretisierung, Auflösung, Empfindlichkeit, Messbereich Signal, Messsignal, Klassifizierung von Signalen (Informationsparameter) Signalbeschreibung, Fourierreihen und Fouriertransformation Fourieranalyse DFT und FFT (praktische Realisierung) Aliasing und Shannon's-Abtasttheorem Übertragungsverhalten (Antwortfunktionen, Frequenzgang, Übertragungsfunktion) Laplace-Transformation, Digitalisierungskette, Z-Transformation und Wavelet-Transformation *Verarbeitung und Übertragung analoger Signale:* Messverstärker, Operationsverstärker (idealer und realer, Rückkopplung) Kenngrößen von Operationsverstärkern Frequenzabhängige Verstärkung von Operationsverstärkern Operationsverstärkertypen Rückkopplung und Grundschaltungen (Komparator, Invertierender Verstärker, Nichtinvertierender Verstärker, Impedanzwandler, Strom-Spannungs-Wandler, Differenzverstärker, Integrierer, Differenzierer, invertierender Addierer, Subtrahierer, Logarithmierer, e-Funktionsgeneratoren, Instrumentenverstärker) OPV mit differentiellen Ausgang analoge Filter (Tiefpassfilter, Hochpassfilter, Bandpassfilter, Bandsperrfilter, Bodeplot, Phasenschiebung, aktive analoge Filter) Messsignalübertragung (Einheitssignale, Anschlussvarianten) Spannungs-Frequenz-Wandler Galvanische Trennung und optische Übertragung Modulatoren und Demodulatoren Multiplexer und Demultiplexer Abtast-Halte-Glied *A/D- und D/A-Umsetzer:* Digitale und analoge Signale Digitalisierungskette A/D-Umsetzer (Nachlauf ADU, Wägeverfahren, Rampen-A/D-Umsetzer, Dual Slope-Verfahren, Charge-Balancing-A/D-Umsetzverfahren, Parallel-A/D-Umsetzer, Kaskaden-A/D-Umsetzverfahren, Pipeline-A/D-Umsetzer, Delta-Sigma-A/D-Umsetzer / 1-Bit- bis N-Bit-Umsetzer, Einsatzbereiche, Kennwerte, Abweichungen, Signal-Rausch-Verhältnis) Digital-Analog-Umsetzungskette D/A-Umsetzer (Direkt bzw. Parallelumsetzer, Wägeumsetzer, Zählverfahren, Pulsweitenmodulation, Delta-Sigma-Umsetzer / 1-Bit- bis N-Bit-Umsetzer) *Verarbeitung digitaler Signale:* digitale Codes Schaltnetze (Kombinatorische Schaltungslogik) Schaltalgebra und logische Grundverknüpfungen Schaltwerke (Sequentielle Schaltnetze) Speicherglieder (Flip-Flops, Sequentielle Grundschaltungen), Halbleiterspeicher (statische und dynamische, FIFO)

		Anwendungsspezifische integrierte Schaltungen (ASICs) Programmierbare logische Schaltung (PLDs, Programmierbarkeit, Vorteile, Anwendungen, Programmierung) Rechnerarten *Bussysteme:* Bussysteme (Master, Slave, Arbiter, Routing, Repeater) Arbitrierung Topologien (physikalische und logische Topologie, Kennwerte, Punkt-zu-Punkt-Topologie, vermaschtes Netz, Stern-Topologie, Ring-Topologie, Bus-Topologie, Baum-Topologie, Zell-Topologie) Übertragungsmedien (Mehrdrahtleitung, Koaxialkabel, Lichtwellenleiter) ISO-OSI-Referenzmodell Physikalische Schnittstellenstandards (RS-232C, RS-422, RS-485) Feldbussysteme, GPIB (IEC-625-Bus), Messgerätebusse *USB Universal Serial Bus:* Struktur des Busses Verbindung der Geräte, Transceiver, Geschwindigkeitserkennung, Signalkodierung Übertragungsarten (Control-Transfer, Bulk-Transfer, Isochrone-Transfer, Interrupt-Transfer, Datenübertragung mit Paketen) Frames und Mikroframes, Geschwindigkeiten, Geschwindigkeitsumsetzung mit Hub Deskriptoren und Software Layer Entwicklungstools Compliance Test USB 3.0 *Digitale Filter:* Analoge Filter Eigenschaften und Charakterisierung von digitalen Filtern Digitale Filter (Implementierung, Topologien, IIR-Filter und FIR-Filter) und Formen Messwert-Dezimierer, digitaler Mittelwertfilter, Gaußfilter Fensterfunktionen, Gibbs-Phänomen Realisierung mit MATLAB Vor- und Nachteile digitaler Filter *Messdatenauswertung:* Absolute, relative, zufällige und systematische Messabweichungen, Umgang mit Messabweichungen, Kalibrierung Korrelationsanalyse Kennlinienabweichungen und Methoden zu deren Ermittlung Regressionsanalyse Kennlinienkorrektur Approximation, Interpolation, Extrapolation Arten der Kennlinienkorrektur Messpräzision, Messgenauigkeit, Messrichtigkeit, Fehlerfortpflanzungsgesetz (altes Konzept), Messunsicherheit und deren Bestimmung Vorgehensweise zur Ermittlung der Unsicherheit, Monte-Carlo-Methode *Schaltungs- und Leiterplattenentwurf:* Leiterplatten Leiterplattenmaterial Leiterplattenarten Durchkontaktierungen Leiterplattenentwurf und -entflechtung Software Leiterplattenherstellung *Contents* *Basics:* Terms (quantity, quantity value, measurand, measurement unit, principle of measurement, measurement, measuring chain, measurement signal, information parameter, analogue and digital signal) Principle of a measuring instrument, direct and indirect measurement, characteristic curves and characteristic curve types, analogue and digital measuring methods, continuous and discontinuous measurement, time and value discretisation, resolution, sensitivity, measuring interval (range) Signal, measurement signal, classification of signals (information parameter) Signal description, Fourier series and Fourier transformation Fourier analysis DFT and FFT (practical realization) Aliasing and Shannon's sampling theorem Transfer behaviour (response functions, frequency response, transfer function) Laplace transform, digitisation chain, Z-transform and wavelet transform

		Processing and transmission of analogue signals: Measuring amplifiers, operational amplifiers (ideal and real, feedback) Characteristics of operational amplifiers Frequency-dependent gain of operational amplifiers Operational amplifier types Feedback and basic circuits (comparator, inverting amplifier, non-inverting amplifier, impedance converter, current-voltage converter, differential amplifier, integrator, differentiator, inverting adder, subtractor, logarithmic, exponential function generators, instrumentation amplifier) OPV with differential output Analogue filter (low pass filter, high pass filter, band pass filter, band elimination filter, Bodeplot, phase shifting, active analogue filters) Measurement signal transmission (standard signals, connection variants) Voltage-frequency converters Galvanic isolation and optical transmission modulators and demodulators multiplexers and demultiplexers sample-and-hold amplifier *A/D and D/A converter:* Digital and analogue signals Digitisation chain A/D converter (follow-up ADC, weighing method, ramp A/D converter, dual slope method, charge-balancing ADC, parallel ADC, cascade ADC, pipeline A/D converter, the delta-sigma A/D converter / 1-bit to N-bit converter, application, characteristics, deviations, signal-to-noise ratio) Digital-to-analogue conversion chain D/A converter (direct or parallel converters, weighing method, counting method, pulse width modulation, delta-sigma converter / 1-bit to N-bit converter) *Digital signal processing:* Digital codes Switching networks (combinatorial circuit logic) Boolean algebra and basic logic operations Sequential circuit (sequential switching networks) Storage elements (flip-flops, sequential basic circuits), semiconductor memory (static and dynamic, FIFO) Application Specific Integrated Circuits (ASICs) The programmable logic device (PLD, programmability, benefits, applications, programming) computer types *Data bus systems:* Bus systems (master, slave, arbiter, routing, repeater) Arbitration Topologies (physical and logical topology, characteristics, point-to-point topology, mesh network, star topology, ring topology, bus topology, tree topology, cell topology) Transmission media (multi-wire cable, coaxial cable, fibre optic cable) ISO OSI reference model Physical interface standards (RS-232C, RS-422, RS-485) Fieldbus systems, GPIB (IEC-625 bus) , Measuring device buses *USB Universal Serial Bus:* Bus structure Connection of the devices, transceiver, speed detection, signal coding Transfer types (control transfer, bulk transfer, isochronous transfer, interrupt transfer, data transfer with packages) Frames and micro-frames, speeds, speed conversion with hubs Descriptors and software Layer development tools Compliance test USB 3.0 *Digital filters:* Analogue filter Properties and characterization of digital filters Digital Filter (implementation, topologies, IIR filters and FIR filters) and forms Measurement value decimator, digital averaging filter, Gaussian filter Window functions, Gibbs phenomenon Realisation with MATLAB Advantages and disadvantages of digital filters *Data analysis:* Absolute, relative, random and systematic errors, handling of measurement errors, calibration Correlation analysis

		Characteristic curve deviations and methods for their determination Regression analysis Characteristic curve correction Approximation, interpolation, extrapolation Kinds of characteristic curve correction Measurement precision, measurement accuracy, measurement trueness, error propagation law (old concept), uncertainty and their estimation Procedure for determining the uncertainty, Monte Carlo method *Circuit and PCB design:* Printed circuit boards (PCB) PCB material PCB types Vias PCB design and deconcentration Software PCB production
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen • Die Studierenden können einen Überblick zur rechnergestützten Messtechnik sowie deren Einsatzgebiete wiedergeben. • Die Studierenden können Wissen zur rechnergestützten Messdatenerfassung, -auswertung, -analyse und visualisierung als Grundlage für zielorientierte, effiziente Entwicklung und für kontinuierliche Produkt- und Prozessverbesserung abrufen Verstehen • ° Die Studierenden können rechnergestützte Werkzeuge für die Messdatenerfassung, -auswertung, -analyse und -visualisierung auswählen und bewerten.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	keine Einpassung in Studienverlaufsplan hinterlegt!
9	**Verwendbarkeit des Moduls**	Wahlpflichtmodule Qualitätsmanagement und Messtechnik für Nebenfach Informatik Bachelor of Science Informatik 20222
10	**Studien- und Prüfungsleistungen**	Klausur (60 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	International Vocabulary of Metrology Basic and General Concepts and Associated Terms, VIM, 3rd edition, JCGM 200:2008, http://www.bipm.org/en/publications/guides/vim.html

		DIN e.V. (Hrsg.): Internationales Wörterbuch der Metrologie Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) ISO/IEC-Leitfaden 99:2007. Korrigierte Fassung 2012, Beuth Verlag GmbH, 4. Auflage 2012 Hoffmann, Jörg: Handbuch der Messtechnik. 4. Auflage, Carl Hanser Verlag München, 2012 ISBN 978-3-446-42736-5 Lerch, Reinhard: Elektrische Messtechnik. 6. Auflage, Springer-Verlag Berlin Heidelberg, 2012 ISBN 978-3-642-22608-3 Richter, Werner: Elektrische Meßtechnik. 3. Auflage, Verlag Technik Berlin, 1994 - ISBN 3-341-01106-4 H. Czichos (Hrsg.): Das Ingenieurwissen Gebundene. 7. Auflage, Springer Verlag, 2012, ISBN 978-3-642-22849-0. Best, Roland: Digitale Meßwertverarbeitung. Oldenbourg München, 1991 - ISBN 3-486-21573-6. E DIN IEC 60050-351:2013-07 International Electrotechnical Vocabulary Part 351: Control technology / Internationales Elektrotechnisches Wörterbuch - Teil 351: Leittechnik. DIN 44300:1982-03 Informationsverarbeitung; Begriffe. DIN 44300-1:1995-03 Informationsverarbeitung - Begriffe - Teil 1: Allgemeine Begriffe. DIN 40900-12:1992-09 Graphische Symbole für Schaltungsunterlagen; Binäre Elemente.