Friedrich-Alexander-Universität
Erlangen-Nürnberg

Modulhandbuch

für den Studiengang

1. Staatsprüfung für das Lehramt an Mittelschulen Informatik

(Prüfungsordnungsversion: 20162)

für das Sommersemester 2025

Inhaltsverzeichnis

Implementierung von Datenbanksystemen (93020)....3
Parallele und Funktionale Programmierung (93040)....6
Algorithmen und Datenstrukturen (93050)....7
Konzeptionelle Modellierung (93130)....11
Theoretische Informatik für Wirtschaftsinformatik und Lehramt (93201)....13
Didaktik der Informatik I (93210)....15
Didaktik der Informatik II (93220)....18
Seminar Didaktik der Informatik (93230)....20
Wahlpflichtbereich Informatik Hauptschule
- Mathematik für Naturwissenschaftler (64640)....22
- Praktikum Informatik in der Bildung (PIB) (93149)....23

Praktikum Informatik
- Project Deep Learning in Multimedia Forensics (47635)....25
- Mobile Application Development and Security (93203)....27
- Rechnerarchitekturen für Deep-Learning Anwendungen (93111)....29
- NWERC Praktikum (93129)....31
- The AMOS Project (SD Role, PROJ 10 ECTS) (93142)....35
- Praktikum Mustererkennung (93155)....37
- Softwareentwicklungspraktikum Lehramt (93162)....39
- Praktikum Künstliche Intelligenz (BSc) (93187)....41
- Hackerpraktikum Bachelor (93192)....42
- Praktikum: Entwicklung interaktiver eingebetteter Systeme (93197)....44
- Product Management (93198)....47
- IoT Security (93199)....48
- Praktikum angewandte Systemsoftwaretechnik (113845)....50
- Grafik-Praktikum Game Programming (240715)....51
- Praktikum Enterprise Computing (594684)....52
- HPC Software Projekt (695344)....53

1	**Modulbezeichnung** 93020	**Implementierung von Datenbanksystemen** Implementation of database 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**	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**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 93040	**Parallele und Funktionale Programmierung** Parallel and functional programming	**5 ECTS**
2	Lehrveranstaltungen	Übung: Intensivübungen zu Parallele und Funktionale Programmierung (2 SWS)	0 ECTS
3	Lehrende	David Schwarzbeck 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**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 93050	**Algorithmen und Datenstrukturen** Algorithms and data structures	**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. Michael Philippsen
5	**Inhalt**	• Grundlagen der Programmierung • Datenstrukturen • Objektorientierung • JAVA-Grundkenntnisse • Aufwandsabschätzungen • Grundlegende Algorithmen
6	**Lernziele und Kompetenzen**	A - Fachkompetenz: 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**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
10	**Studien- und Prüfungsleistungen**	Klausur (120 Minuten) Übungsleistung

11	**Berechnung der Modulnote**	Klausur (100%) Übungsleistung (bestanden/nicht bestanden)
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** 93130	**Konzeptionelle Modellierung** Conceptual modelling	**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.-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**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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**	• 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** 93201	**Theoretische Informatik für Wirtschaftsinformatik und Lehramt** Theoretical computer science for information systems and teaching degree students	**5 ECTS**
2	Lehrveranstaltungen	Übung: ÜThInfWI-A (2 SWS) Vorlesung: Theoretische Informatik für Wirtschaftsinformatik und Lehramt (2 SWS) Übung: Intensivübung zu Theoretische Informatik für Wirtschaftsinformatik und Lehramt (optional) (2 SWS)	- - -
3	Lehrende	Simon Prucker Max Ole Elliger apl. Prof. Dr. Stefan Milius	

4	**Modulverantwortliche/r**	apl. Prof. Dr. Stefan Milius
5	**Inhalt**	Grundlegende Begriffe und Kernergebnisse der Automatentheorie, Berechenbarkeitstheorie und Komplexitätstheorie werden überblickhaft behandelt: • endliche Automaten und reguläre Grammatiken und Sprachen • Kellerautomaten, kontextfreue Grammatiken und Sprachen • Turingmaschinen und berechenbare Funktionen • Primitiv rekursive und mü-rekursive Funktionen • LOOP- und WHILE-Berechenbarkeit • Entscheidbare Sprachen und Unentscheidbarkeit • Chomsky-Hierarchie • Komplexitätsklassen P und NP • NP-Vollständigkeit
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben elementare Definitionen und Fakten zu formalen Sprachen und entsprechenden Maschinenmodellen und Grammatiken wieder. Verstehen Die Studierenden • erklären grundlegende Konzepte der Begriffe der Automaten- und Berechenbarkeits- und Komplexitätstheorie. • beschreiben Beispiele dieser Konzepte. • erläutern grundlegende Konstruktionen, Algorithmen und wesentliche Resultate und entsprechende Beweise (z.B. Unentscheibarkeit des Halteproblems). Anwenden Die Studierenden • führen Konstruktionen auf vorgelegten Maschinen und Grammatiken und Sprachen durch (z.B. Automatenminiierung, Potenzmengen-Konstruktion, Chomsky-Normierung, CYK-Algorithmus). • wenden grundlegende Beweisverfahren der theoretischen Informatik an (z.B. Induktionsbeweise, Pumping-Lemma, Reduktionen).

		Analysieren Die Studierenden • analysieren formale Sprachen und ermitteln ihre Zugehörigkeit zu den Klassen der Chomsky-Hierarchie. • untersuchen die Entscheidbarkeit von vorgelegten formalen Sprachen. • analysieren die Komplexität eine Entscheidungsproblem und klassifizieren es als Problem in P, NP bzw. NP-vollständig. 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: 6
9	**Verwendbarkeit des Moduls**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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: 56 h Eigenstudium: 94 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• U. Schöning: Theoretische Informatik - kurz gefasst, 5. Aufl., Spektrum 2008. • J.E. Hopcroft, R. Motwani und J.D. Ullman: Introduction to Automata Theory, Languages, and Computation, 2. Aufl., Addison Wesley, 2001.

1	**Modulbezeichnung** 93210	**Didaktik der Informatik I** Teaching computer science I	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Didaktik der Informatik I (4 SWS)	5 ECTS
3	Lehrende	Prof. Dr. Marc-Pascal Berges	

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**	Module Fachdidaktik Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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	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-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: 5
9	**Verwendbarkeit des Moduls**	Module Fachdidaktik Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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 (bestanden/nicht bestanden) 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** 93230	**Seminar Didaktik der Informatik** Seminar: Teaching computer science	**2,5 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-Pascal Berges
5	**Inhalt**	• Anwendungen informatikbezogener Lehr-Lernprozesse • Unterrichtsentwicklung mit Werkzeugen für informatische Bildungsgegenstände • Methoden und Techniken des Lehrens und Lernens in formellen und informellen Kontexten der Informatik
6	**Lernziele und Kompetenzen**	Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, informatische Bildungsprozesse an ausgewählten Beispielen auf wissenschaftlicher Grundlage zu gestalten und zu reflektieren. Sie • diskutieren und erproben ausgewählte Forschungsansätze der Didaktik der Informatik • entwerfen Beispiele im Gegenstandsbereich • entwickeln und evaluieren informatische Lehr-Lernprozesse unter Berücksichtigung des angestrebten Lehramtsabschlusses
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Module Fachdidaktik Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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: 45 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

Wahlpflichtbereich Informatik Hauptschule

1	**Modulbezeichnung** 64640	**Mathematik für Naturwissenschaftler** Mathematics for natural scientists	**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. Alexander Prechtel
5	**Inhalt**	• Grundbegriffe der linearen Algebra und Analysis • Komplexe Zahlen • Lineare Abbildungen, Matrizen, Gauss-Algorithmus, Determinanten, Eigenwerte und Eigenvektoren, Diagonalisierung • Stetige und differenzierbaren Funktionen, Taylor-Reihen, Integralrechung • Stabilitätsanalyse linearer Differentialgleichungssysteme 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 • definieren und erklären Grundbegriffe der Analysis und linearen Algebra; • verwenden grundlegende Verfahren und Algorithmen; • diskutieren Funktionen, Folgen und Reihen; • sammeln relevante Informationen, erkennen Zusammenhänge und bewerten diese.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 6
9	**Verwendbarkeit des Moduls**	Wahlpflichtbereich Informatik Hauptschule 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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**	Sämtliche Literatur mit Titel "Mathematik für Chemiker" oder "Ingenieursmathematik".

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: 6
9	**Verwendbarkeit des Moduls**	Wahlpflichtbereich Informatik Hauptschule 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
10	**Studien- und Prüfungsleistungen**	Variabel Die Prüfungsleistung besteht aus einem Praktikumsbericht in Form eines Produktportfolios. Der Umfang beträgt 50-100 Seiten. Der Praktikumsbericht muss zum Abschluss in einem Vortrag von 20min. Länge vorgestellt werden.
11	**Berechnung der Modulnote**	Variabel (100%) Die Modulnote ergibt sich aus der Bewertung des Praktikumsberichts.
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**	

Praktikum Informatik

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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162

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** 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**	Semester: 7

9	**Verwendbarkeit des Moduls**	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 93111	**Rechnerarchitekturen für Deep-Learning Anwendungen** Computer architectures for deep-learning applications	**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. 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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 93129	**NWERC Praktikum** NWERC Training	**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. 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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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	Vorlesung: The AMOS Project (VL) Übung: The AMOS Project (UE)	- -
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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 93155	**Praktikum Mustererkennung** Laboratory course: Pattern recognition	**10 ECTS**
2	Lehrveranstaltungen	Praktikum: Praktikum Mustererkennung (4 SWS)	5 ECTS
3	Lehrende	Prof. Dr.-Ing. Andreas Maier	

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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 93162	**Softwareentwicklungspraktikum Lehramt** Software development lab for teaching degree students	**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. 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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 93187	**Praktikum Künstliche Intelligenz (BSc)** Laboratory course: Artificial intelligence (BSc)	**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. 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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 93192	**Hackerpraktikum Bachelor** Hacking lab (Bachelor´s students)	**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
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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 93197	**Praktikum: Entwicklung interaktiver eingebetteter Systeme** Laboratory course: Development of cyber-physical embedded 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**	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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 240715	**Grafik-Praktikum Game Programming** Laboratory course: Games programming (GraPra)	**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**	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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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** 695344	**HPC Software Projekt** HPC Software project	**10 ECTS**
2	Lehrveranstaltungen	Praktikum/Projekt: HPC Software Projekt (8 SWS)	10 ECTS
3	Lehrende	apl. Prof. Dr. Harald Köstler	

4	**Modulverantwortliche/r**	apl. 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 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20162
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**	