

Modulhandbuch

für den Studiengang

1. Staatsprüfung für das Lehramt
an Gymnasien Informatik
(Prüfungsordnungsversion: 2007)

Inhaltsverzeichnis

Algorithmen und Datenstrukturen (93050).....	4
Didaktik der Informatik I (93210).....	8
Didaktik der Informatik II (93220).....	11
Grundlagen der Technischen Informatik (93110).....	13
Implementierung von Datenbanksystemen (93020).....	15
Konzeptionelle Modellierung (93130).....	18
Parallele und Funktionale Programmierung (93040).....	20
Rechnerkommunikation (93150).....	21
Software-Entwicklung in Großprojekten (93160).....	23
Systemprogrammierung (93180).....	25
Theoretische Informatik für Wirtschaftsinformatik und Lehramt (93201).....	27
Wahlpflichtbereich Informatik Gymnasium	
Advanced Mechanized Reasoning in Coq (93169).....	30
Algebra des Programmierens (861501).....	31
Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML) (510375).....	33
Angewandte IT-Sicherheit (326311).....	35
Begleitseminar zur schriftlichen Hausarbeit (819845).....	37
Berechenbarkeit und Formale Sprachen (93010).....	38
Computergraphik (Vorlesung mit Übung und Praktikum) (43394).....	40
CPU Entwurf mit VHDL (CPU) (211243).....	43
Datenbanken in Rechnernetzen und Transaktionssysteme (681735).....	45
Didaktik der Informatik in der Schulpraxis (93081).....	48
Einführung in die Medizinische Informatik (93300).....	50
Forensische Informatik (792501).....	52
Grundlagen der Logik in der Informatik (93072).....	54
Grundlagen der Rechnerarchitektur und -organisation (93080).....	57
Grundlagen der Schaltungstechnik (93091).....	59
Grundlagen des Software Engineering (93550).....	61
Hardware-Software-Co-Design (502509).....	63
Human Computer Interaction (645618).....	66
Human Computer Interaction (645618).....	69
Introduction to Machine Learning (65718).....	72
Knowledge Discovery in Databases mit Übung (93107).....	75
Künstliche Intelligenz I (894856).....	77
Künstliche Intelligenz II (532733).....	79
Logik-Basierte Sprachverarbeitung (675137).....	81
Mathematik für Naturwissenschaften (64640).....	83
Mathematische Modellbildung und Statistik für Naturwissenschaftler (65760).....	85
Monad-Based Programming (845618).....	87
Praktikum Informatik in der Bildung (PIB) (93149).....	90
Praktische Semantik von Programmiersprachen (599478).....	91
Scientific Visualization (43722).....	93
Seminar Didaktik der Informatik (93230).....	95
Seminar 'Hallo Welt!' für Fortgeschrittene (319374).....	96
Seminar Informatik in der Bildung (93163).....	98
Themen der Kategorientheorie (819238).....	99
Theorie der Programmierung (93121).....	101
Visualization (93175).....	103
Praktikum Informatik	

Grafik-Praktikum Game Programming (240715).....	106
HPC Software Projekt (695344).....	107
IoT Security (93199).....	109
Mobile Application Development and Security (93203).....	111
NWERC Praktikum (93129).....	113
The AMOS Project (SD Role, PROJ 10 ECTS) (93142).....	117
Praktikum: Entwicklung interaktiver eingebetteter Systeme (93197).....	119
Praktikum Lego Mindstorms (278855).....	122
Praktikum Mustererkennung (93155).....	124
Product Management (93198).....	126
Softwareentwicklungspraktikum Lehramt (93162).....	127

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	<ul style="list-style-type: none"> • Grundlagen der Programmierung • Datenstrukturen • Objektorientierung • JAVA-Grundkenntnisse • Aufwandsabschätzungen • Grundlegende Algorithmen 	
6	Lernziele und Kompetenzen	<p><u>A - Fachkompetenz:</u> Die Studierenden...</p> <p>1.) Grundlagen der Programmierung in Java</p> <ul style="list-style-type: none"> • 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 -Wertzuzuweisung 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 <p>2.) Rekursion</p> <ul style="list-style-type: none"> • 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
- instantiiieren 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 verschiedenen 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)

		<ul style="list-style-type: none"> • 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 <p>7.) Bäume und Graphen</p> <ul style="list-style-type: none"> • 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 <p><u>B - Selbst- und Sozialkompetenz:</u> Die Studierenden...</p> <ul style="list-style-type: none"> • 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	<p>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.</p>
8	Einpassung in Studienverlaufsplan	Semester: 1
9	Verwendbarkeit des Moduls	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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 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	<ul style="list-style-type: none"> • 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	<p>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.</p> <p>Sie</p> <ul style="list-style-type: none"> • 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

		<p>begründen die Relevanz informatischen Modellierens für die Schulinformatik</p> <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Modul 3050 Algorithmen und Datenstrukturen • Modul 3200 Theoretische Informatik für Lehramtsstudierende

		<ul style="list-style-type: none"> • Modul 3130 Konzeptionelle Modellierung
8	Einpassung in Studienverlaufsplan	Semester: 3
9	Verwendbarkeit des Moduls	Module Fachdidaktik Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Theoretische Fundierung der Didaktik der Informatik • Voraussetzungen und Rahmenbedingungen für Informatikunterricht • Lern- und Kompetenzziele des Informatikunterrichts 	
6	Lernziele und Kompetenzen	<p>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.</p>	
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 Gymnasien Informatik 2007 Module Fachdidaktik Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2022	
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	<ul style="list-style-type: none"> • Hubwieser, Peter. Didaktik der Informatik. Springer-Verlag, 2007. • Schubert, Sigrid, and Andreas Schwill. Didaktik der Informatik. Spektrum Akademischer Verlag, 2011.

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

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, Zustandskodierung 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	<p>Fachkompetenz - Wissen</p> <ul style="list-style-type: none"> 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. <p>Fachkompetenz - Verstehen</p> <ul style="list-style-type: none"> Die Studierenden verstehen, dass Hardware heutzutage mit Software am Rechner entwickelt und simuliert wird. Die Studierenden verstehen den Schaltungsentwurf mittels einer Beschreibungssprache (VHDL). <p>Fachkompetenz - Anwenden</p> <ul style="list-style-type: none"> Die Studierenden erarbeiten und diskutieren verschiedene Lösungswege für die Datencodierung sowie den Entwurf und die Optimierung von digitalen Hardwareschaltungen. <p>Selbstkompetenz</p>	

		<ul style="list-style-type: none"> 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: 5
9	Verwendbarkeit des Moduls	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<p>Weitere Informationen:</p> <p>https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/vorlesungen/grundlagen-der-technischen-informatik-im-wintersemester</p>

1	Modulbezeichnung 93020	Implementierung von Datenbanksystemen Implementation of database systems	5 ECTS
2	Lehrveranstaltungen	Vorlesung: Implementierung von Datenbanksystemen (2 SWS) Übung: IDB ÜG3 (2 SWS) Übung: IÜ (2 SWS)	2,5 ECTS 2,5 ECTS -
3	Lehrende	Demian Vöhringer Konstantin Müller Felix Hanika	

4	Modulverantwortliche/r	Prof. Dr.-Ing. Richard Lenz
5	Inhalt	<p>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.</p> <p>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.</p>
6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <ul style="list-style-type: none"> • 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;

		<ul style="list-style-type: none"> • 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 Gymnasien Informatik 2007
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	<p>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</p> <p>KEMPER, Alfons ; WIMMER, Martin: Übungsbuch Datenbanksysteme. 2., aktual. u. erweit. Aufl. München : Oldenbourg, 2009. ISBN 978-3-486-59001-2. Kapitel 7 bis 11</p> <p>HEUER, Andreas ; SAAKE, Gunter: Datenbanken : Konzepte und Sprachen. 3., aktual. u. erw. Aufl. Bonn : mitp, 2007. - ISBN 3-8266-1664-2</p> <p>HÄRDER, Theo ; RAHM, Erhard: Datenbanksysteme : Konzepte und Techniken der Implementierung. Berlin : Springer, 1999 - ISBN 3-540-65040-7</p> <p>SAAKE, Gunter ; HEUER, Andreas: Datenbanken : Implementierungstechniken. 2., aktual. u. erw. Aufl. Bonn : mitp, 2005. ISBN 3-8266-1438-0</p>

1	Modulbezeichnung 93130	Konzeptionelle Modellierung Conceptual modelling	5 ECTS
2	Lehrveranstaltungen	Übung: WS23-02 (2 SWS) Vorlesung: Konzeptionelle Modellierung (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	David Haller Constantin Winkler Dominik Probst Alexander Seifert Prof. Dr.-Ing. Richard Lenz	

4	Modulverantwortliche/r	Prof. Dr.-Ing. Richard Lenz	
5	Inhalt	<ul style="list-style-type: none"> • 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	<p>Die Studierenden:</p> <ul style="list-style-type: none"> • 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 	

		<ul style="list-style-type: none"> • 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: 3
9	Verwendbarkeit des Moduls	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<ul style="list-style-type: none"> • 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 93040	Parallele und Funktionale Programmierung Parallel and functional programming	5 ECTS
2	Lehrveranstaltungen	Übung: PFP-R-Di16 (2 SWS) Vorlesung: Parallele und Funktionale Programmierung (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Julian Brandner Prof. Dr. Michael Philippsen Dr.-Ing. Norbert Oster	

4	Modulverantwortliche/r	Dr.-Ing. Norbert Oster Prof. Dr. Michael Philippsen	
5	Inhalt	<ul style="list-style-type: none"> • Grundlagen der funktionale Programmierung • Grundlagen der parallelen Programmierung • Datenstrukturen • Objektorientierung • Scala-Kenntnisse • Erweiterte JAVA-Kenntnisse • Aufwandsabschätzungen • Grundlegende Algorithmen 	
6	Lernziele und Kompetenzen	Die Studierenden <ul style="list-style-type: none"> • 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 Gymnasien 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	<p>Die Vorlesung vermittelt die Grundlagen der Rechnerkommunikation und durchläuft von oben nach unten die Schichten des Internets:</p> <ul style="list-style-type: none"> • Anwendungsschicht • Transportschicht • Netzwerkschicht • Sicherungsschicht • Physikalische Schicht <p>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.</p>	
6	Lernziele und Kompetenzen	<p>Die Studierenden erwerben</p> <ul style="list-style-type: none"> • 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;4	
9	Verwendbarkeit des Moduls	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20222	
10	Studien- und Prüfungsleistungen	<p>Übungsleistung Klausur (90 Minuten) Hausaufgaben zu Rechnerkommunikation (Übungsleistung):</p> <ul style="list-style-type: none"> • 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 <p>Rechnerkommunikation (Klausur):</p> <ul style="list-style-type: none"> • 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 93160	Software-Entwicklung in Großprojekten Software development in large projects	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. Francesca Saglietti	
5	Inhalt	<ul style="list-style-type: none"> • 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	<p>Die Studierenden</p> <ul style="list-style-type: none"> • 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	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien 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	Modulbezeichnung 93180	Systemprogrammierung System programming	10 ECTS
2	Lehrveranstaltungen	Vorlesung: Systemprogrammierung 2 (2 SWS) Übung: SP-RÜ R02, Christian Halder (2 SWS) Übung: Systemprogrammierung 1 - Übung (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 Maximilian Ott Thomas Preisner Jonas Rabenstein	

4	Modulverantwortliche/r	Dr.-Ing. Jürgen Kleinöder Prof. Dr.-Ing. Wolfgang Schröder-Preikschat
5	Inhalt	<ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • 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 Systemaufrufchnittstelle UNIX-ähnlicher Betriebssysteme
7	Voraussetzungen für die Teilnahme	Keine
8	Einpassung in Studienverlaufsplan	Semester: 4
9	Verwendbarkeit des Moduls	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<ul style="list-style-type: none"> Lehrbuch: Betriebssysteme Grundlagen, Entwurf, Implementierung, Wolfgang Schröder-Preikschat, 2008

1	Modulbezeichnung 93201	Theoretische Informatik für Wirtschaftsinformatik und Lehramt Theoretical computer science for information systems and teaching degree students	5 ECTS
2	Lehrveranstaltungen	Die Lehrveranstaltungen des Moduls werden nur im Sommersemester angeboten.	
3	Lehrende	-	

4	Modulverantwortliche/r	apl.Prof.Dr. Stefan Milius	
5	Inhalt	<p>Grundlegende Begriffe und Kernergebnisse der Automatentheorie, Berechenbarkeitstheorie und Komplexitätstheorie werden überblickhaft behandelt:</p> <ul style="list-style-type: none"> • endliche Automaten und reguläre Grammatiken und Sprachen • Kellerautomaten, kontextfreie 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	<p>Fachkompetenz Wissen Die Studierenden geben elementare Definitionen und Fakten zu formalen Sprachen und entsprechenden Maschinenmodellen und Grammatiken wieder.</p> <p>Verstehen Die Studierenden</p> <ul style="list-style-type: none"> • 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. Unentscheidbarkeit des Halteproblems). <p>Anwenden Die Studierenden</p> <ul style="list-style-type: none"> • führen Konstruktionen auf vorgelegten Maschinen und Grammatiken und Sprachen durch (z.B. Automatenminimierung, Potenzmengen-Konstruktion, Chomsky-Normierung, CYK-Algorithmus). • wenden grundlegende Beweisverfahren der theoretischen Informatik an (z.B. Induktionsbeweise, Pumping-Lemma, Reduktionen). <p>Analysieren Die Studierenden</p> <ul style="list-style-type: none"> • analysieren formale Sprachen und ermitteln ihre Zugehörigkeit zu den Klassen der Chomsky-Hierarchie. 	

		<ul style="list-style-type: none"> • untersuchen die Entscheidbarkeit von vorgelegten formalen Sprachen. • analysieren die Komplexität eines Entscheidungsproblems und klassifizieren es als Problem in P, NP bzw. NP-vollständig. <p>Lern- bzw. Methodenkompetenz Die Studierenden</p> <ul style="list-style-type: none"> • 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. <p>Sozialkompetenz Die Studierenden lösen Probleme in kollaborativer Gruppenarbeit und präsentieren erarbeitete Lösungen.</p>
7	Voraussetzungen für die Teilnahme	Keine
8	Einpassung in Studienverlaufsplan	Semester: 2
9	Verwendbarkeit des Moduls	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien 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: 56 h Eigenstudium: 94 h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	Deutsch
16	Literaturhinweise	<ul style="list-style-type: none"> • 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.

Wahlpflichtbereich Informatik Gymnasium

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: 1
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<p>Induktive Datentypen wie z.B. Listen, Stacks und Bäume werden abstrakt beschrieben.</p> <p>Strukturelle Induktion und Rekursion für solche Datentypen (z.B. die fold-Operation auf Listen) werden auf Grundlage der Initiale-Algebra-Semantik entwickelt.</p> <p>Dadurch werden verschiedene effektive Programmiertricks auf eine solide mathematische Grundlage gestellt.</p> <p>Grundlagen und Methoden der Kategorientheorie werden eingeführt und erklärt, insbesondere initiale Algebren und ihre Konstruktion.</p> <p>Evtl. werden Koalgebren behandelt, die es ermöglichen, verschiedene zustandsbasierte Systeme und ihre Semantik in einer einheitlichen Theorie zu studieren.</p>	
6	Lernziele und Kompetenzen	<p>Fachkompetenz Wissen Die Studierenden geben elementare Definitionen und Fakten zu induktiven Datentypen, zustandsbasierten Systemen und grundlegenden kategoriellen Begriffen wieder</p> <p>Verstehen Die Studierenden</p> <ul style="list-style-type: none"> • erklären grundlegende Begriffe und Konzepte der Kategorientheorie • beschreiben Beispiele dieser Begriffe und Konzepte • erläutern grundlegende kategorielle Ergebnisse und deren Beweise <p>Anwenden Die Studierenden</p> <ul style="list-style-type: none"> • wenden strukturelle Induktionsprinzipien (Definitions- und Beweisprinzip) auf Programmierbeispiele an • wenden strukturelle Koinduktionsprinzipien (Definitions- und Beweisprinzip) auf festgelegte Typen von zustandsbasierten Systemen (Koalgebren) an • übertragen kategorielle Begriffe auf verschiedene Spezialfälle der Informatik (abstrakte Datentypen und verschiedene Arten von Automaten) <p>Analysieren Die Studierenden analysieren kategorielle Beweise, diskutieren die entsprechenden Argumentationen und legen diese schriftlich klar nieder.</p> <p>Erschaffen Die Studierenden</p>	

		<ul style="list-style-type: none"> • 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 <p>Lern- bzw. Methodenkompetenz Die Studierenden</p> <ul style="list-style-type: none"> • beherrschen das grundsätzliche Konzept des Beweises als hauptsächlich 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 <p>Sozialkompetenz Die Studierenden lösen Probleme in kollaborativer Gruppenarbeit und präsentieren erarbeitete Lösungen.</p>
7	Voraussetzungen für die Teilnahme	Keine
8	Einpassung in Studienverlaufsplan	Semester: 6
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<ul style="list-style-type: none"> • 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 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	<p>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.</p> <p>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.</p>	
6	Lernziele und Kompetenzen	<p>Im Rahmen dieser Veranstaltung sollen die Studierenden insbesondere die Kompetenz erwerben,</p> <ul style="list-style-type: none"> - 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: 6
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien 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	<ul style="list-style-type: none"> • 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 <p>Die einschlägige Originalliteratur zur UML findet man auf der Website der Object Management Group (http://www.omg.org/spec/UML).</p>

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	<p>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.</p> <p>Wichtiger Hinweis:</p> <p>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).</p> <p>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.</p> <p>Link zum StudON-Kurs: https://www.studon.fau.de/crs4774802.html</p>	
6	Lernziele und Kompetenzen	<p>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.</p>	
7	Voraussetzungen für die Teilnahme	Keine	

8	Einpassung in Studienverlaufsplan	keine Einpassung in Studienverlaufsplan hinterlegt!
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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 819845	Begleitseminar zur schriftlichen Hausarbeit Accompanying seminar for written assignment	2,5 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-Pascal Berges	
5	Inhalt	Forschungsthemen und methoden der Didaktik der Informatik.	
6	Lernziele und Kompetenzen	Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, Teilaspekte informatikdidaktischer Forschung eigenständig wissenschaftlich adäquat zu bearbeiten und zu präsentieren.	
7	Voraussetzungen für die Teilnahme	Durchgeführte Anmeldung zur schriftlichen Hausarbeit im Lehramtsstudiengang für das Lehramt an Realschulen, Gymnasien oder beruflichen Schulen.	
8	Einpassung in Studienverlaufsplan	Semester: 6	
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007	
10	Studien- und Prüfungsleistungen	mündlich	
11	Berechnung der Modulnote	mündlich (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 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 Matthias Kergaßner	

4	Modulverantwortliche/r	Prof. Dr. Rolf Wanka	
5	Inhalt	<ul style="list-style-type: none"> • 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	<p>Die Studierenden</p> <ul style="list-style-type: none"> • 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: 7	
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien 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	<ul style="list-style-type: none">• I. Wegener. Theoretische Informatik.• J. Hopcroft, J. Ullman. Introduction to Automata Theory, Languages and Computation.• U. Schöning. Theoretische Informatik - kurz gefasst.

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 Prof. Dr. Marc Stamminger Laura Fink	

4	Modulverantwortliche/r	Prof. Dr. Marc Stamminger	
5	Inhalt	<p>Die Vorlesung gibt eine Einführung in die Computergraphik:</p> <ul style="list-style-type: none"> • 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	<p>Die Studierenden</p> <ul style="list-style-type: none"> • 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 	

		<ul style="list-style-type: none"> • klassifizieren Schattierungsverfahren • bestimmen den Unterschied zwischen lokaler und globaler Beleuchtung und formulieren Algorithmen für Ray Tracing und Radiosity <p>*Educational objectives and skills:*</p> <p>Students should be able to</p> <ul style="list-style-type: none"> • 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: 7
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien 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	<ul style="list-style-type: none"> • P. Shirley: Fundamentals of Computer Graphics. AK Peters Ltd., 2002 • Hearn, M. P. Baker: Computer Graphics with OpenGL. Pearson • Foley, van Dam, Feiner, Hughes: Computer Graphics - Principles and Practice

- Rauber: Algorithmen der Computergraphik
- Bungartz, Griebel, Zenger: Einführung in die Computergraphik
- Encarnaç o, Strasser, Klein: Computer Graphics

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	<p>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?</p> <p>Um die Vorlesung verstehen und die Übungsaufgaben lösen zu können, werden VHDL-Kenntnisse benötigt. Diese werden in der Vorlesung vermittelt.</p> <p>Im ersten Teil der Vorlesung geht es darum, aus einfachen Und-, Oder- und Inverter-Gattern größere Strukturen aufzubauen. So werden beispielsweise 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.</p> <p>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.</p>
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: 6
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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 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	<p>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.</p> <p>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:</p> <ul style="list-style-type: none"> • 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 <p>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.</p>	
6	Lernziele und Kompetenzen	<p>DBRN: Die Studierenden</p> <ul style="list-style-type: none"> • 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

		<ul style="list-style-type: none"> • 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: 1
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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 93081	Didaktik der Informatik in der Schulpraxis Teaching computer science in schools	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-Pascal Berges	
5	Inhalt	Die Studierenden erstellen ausführliche, kommentierte und fachlich angereicherte Lösungsvorschläge zu Aufgaben des Staatsexamens in Informatik aus den vorangegangenen Jahren. Sie arbeiten dabei in Eigenregie und in Absprache mit anderen Studierenden. Dabei erlangen Sie ein tieferes Verständnis der zugrunde liegenden Prüfungsinhalte des Staatsexamens und erarbeiten sich angemessene Techniken zur Prüfungsbearbeitung.	
6	Lernziele und Kompetenzen	Die Studierenden sind in der Lage, Aufgaben aus den Bereichen Algorithmen und Datenstrukturen, Theoretische Informatik, Datenbanksysteme, Softwaredidaktik und Informatikdidaktik auf Staatsexamensniveau angemessen und fachlich richtig zu bearbeiten. Sie können die Anforderungen von Staatsexamensaufgaben analysieren und daraus die erforderlichen Bearbeitungsschritte und das erforderliche formale Vorgehen ableiten.	
7	Voraussetzungen für die Teilnahme	Es wird der erfolgreiche Abschluss aller Pflichtmodule dringend empfohlen, insbesondere: Didaktik der Informatik 1-3 Einführung in die Programmierung Einführung in die Algorithmen Einführung in Datenbanken Einführung in das Software Engineering Theoretische Informatik für Wirtschaftsinformatik und Lehramt	
8	Einpassung in Studienverlaufsplan	Semester: 9;7	
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007	
10	Studien- und Prüfungsleistungen	Seminarleistung Seminarleistung: Mindestens zwei Ausarbeitungen von Staatsexamensaufgaben und Vorstellung der Ergebnisse in einem dem Aufgabenumfang angemessenen Vortrag, darin auch kritische Reflexion der Schwierigkeiten der Aufgaben und Erschließung von im Aufgabenkontext vorkommenden Thematiken. Zusätzlich wird die regelmäßige Teilnahme am Seminar zur Diskussion und Reflexion der Erarbeitungen anderer Teilnehmender erwartet.	
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	
16	Literaturhinweise	

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	<p>Die Vorlesung soll an medizinischer Informatik interessierten Studenten einen groben thematischen Überblick über verschiedene Schwerpunkte der Medizinischen Informatik geben. Vorgestellte Themenbereiche:</p> <ul style="list-style-type: none"> • Informationssysteme im Gesundheitswesen • Medizinische Biometrie und Epidemiologie • mHealth • Bioinformatik 	
6	Lernziele und Kompetenzen	<p>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.</p> <p>Fachkompetenz Wissen</p> <ul style="list-style-type: none"> • nennen die Grundbegriffe im Bereich Epidemiologie und Biometrie • nennen grundlegende Werkzeuge und Fragestellungen der Bioinformatik <p>Verstehen</p> <ul style="list-style-type: none"> • 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 	

		<ul style="list-style-type: none"> • 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 <p>Anwenden</p> <ul style="list-style-type: none"> • 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 <p>Analysieren</p> <ul style="list-style-type: none"> • analysieren Kontexterkenkung, insbesondere in der isolierten Klassifikation von Aktivitäten anhand von inertialen Sensordaten aus körpergetragenen / Wearable Systemen
7	Voraussetzungen für die Teilnahme	Keine
8	Einpassung in Studienverlaufsplan	Semester: 6
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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 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	<p>Forensische Informatik befasst sich mit der Sammlung, Aufbereitung und Analyse digitaler Beweismittel zur Verwendung vor Gericht. Ausgangspunkt ist jeweils der Verdacht auf einen Computereintrich oder eine Straftat, die mit Hilfe von digitalen Geräten vorgenommen worden ist.</p> <p>Dieses Modul gibt einen Überblick über die Methoden der forensischen Informatik aus einer wissenschaftlichen Perspektive.</p> <p>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.</p> <p>Voraussichtliche Themen:</p> <ul style="list-style-type: none"> • 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	<p>Die Studierenden können Termini und Methoden der digitalen Forensik in die Entwicklung der forensischen Wissenschaften einordnen.</p> <p>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.</p> <p>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.</p>	
7	Voraussetzungen für die Teilnahme	Keine	
8	Einpassung in Studienverlaufsplan	Semester: 6	
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien 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	<ul style="list-style-type: none"> • 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 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: <ul style="list-style-type: none"> • Syntax und Semantik • Automatisches Schließen: Resolution • Formale Deduktion: Korrektheit, Vollständigkeit Prädikatenlogik erster Stufe: <ul style="list-style-type: none"> • Syntax und Semantik • Automatisches Schließen: Unifikation, Resolution • Quantorenelimination • Anwendung automatischer Beweiser • Formale Deduktion: Korrektheit, Vollständigkeit
6	Lernziele und Kompetenzen	<ul style="list-style-type: none"> • 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 Problemspezifikationen (Kohärenz, Widerspruchsfreiheit) und ihre Umsetzung in Logikprogramme; • Beherrschung der praktischen Aspekte der Logikprogrammierung. Fachkompetenz Wissen Die Studierenden

		<p>geben Definitionen zur Syntax und Semantik der verwendeten Logiken wieder</p> <p>beschreiben grundlegende Deduktionsalgorithmen</p> <p>geben Regeln der verwendeten formalen Deduktionssysteme wieder</p> <p>Verstehen</p> <p>Die Studierenden</p> <p>erläutern das Verhältnis zwischen Syntax, Semantik und Beweistheorie der verwendeten Logiken</p> <p>erklären die Funktionsprinzipien grundlegender Deduktionsalgorithmen</p> <p>erläutern die Funktionsweise automatischer Beweiser</p> <p>erläutern grundlegende Resultate der Metatheorie der verwendeten Logiken und deren Bedeutung</p> <p>Anwenden</p> <p>Die Studierenden</p> <p>wenden Deduktionsalgorithmen auf konkrete Deduktionsprobleme an</p> <p>formalisieren Anwendungsprobleme in logischer Form und verwenden automatische Beweiser zur Erledigung entstehender Beweisziele</p> <p>führen einfache formale Beweise manuell</p> <p>Analysieren</p> <p>Die Studierenden führen einfache metatheoretische Beweise, insbesondere durch syntaktische Induktion</p> <p>Lern- bzw. Methodenkompetenz</p> <p>Die Studierenden beherrschen das grundsätzliche Konzept des Beweises als hauptsächliche Methode des Erkenntnisgewinns in der theoretischen Informatik. Sie überblicken abstrakte Begriffsarchitekturen.</p> <p>Sozialkompetenz</p> <p>Die Studierenden lösen abstrakte Probleme in Gruppenarbeit.</p>
7	Voraussetzungen für die Teilnahme	Keine
8	Einpassung in Studienverlaufsplan	Semester: 7
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien 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	<p>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 Ablaufsteuerung bei der Verarbeitung 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.</p> <p>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.</p>	
6	Lernziele und Kompetenzen	<p>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 Computersystems einordnen. Sie kennen den Aufbau von Caches, bzw. von Speichern im Allgemeinen und verstehen die Funktionsweise der Ablaufsteuerung, 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.</p>	
7	Voraussetzungen für die Teilnahme	Keine	

8	Einpassung in Studienverlaufsplan	Semester: 6
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien 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	<p>Hennessy/Patterson: Computer Architecture - A quantitative approach, 4.Auflage, 2006, MorganKaufmann.</p> <p>Patterson/Hennessy: Computer Organization & Design, 4.Auflage, 2008, MorganKaufmann.</p> <p>Stallings, Computer Organization & Architecture, 8.Auflage, 2009, Prentice Hall.</p> <p>Märting, Rechnerarchitekturen, 2001, Fachbuchverlag Leipzig.</p>

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	<p>Im Modul werden grundlegende Zusammenhänge elektrotechnischer Systeme, welche die Grundlage fast aller digitalen Datenverarbeitungssystemen bilden, behandelt. Zu Beginn werden elektrotechnische 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 Grundsaltungen 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.</p> <p>Elektrotechnische Grundlagen</p> <ul style="list-style-type: none"> • Mathematische Grundlagen • Netzwerkanalyse Gleichstromfall • Netzwerkanalyse Wechselstromfall <p>Elektronische Bauelemente</p> <ul style="list-style-type: none"> • Widerstand, Kondensator, Spule (passiv) • Diode, Transistor, Operationsverstärker (aktiv) <p>Einfache dynamische Vorgänge in Schaltungen (Schaltvorgänge)</p> <p>Digitale Schaltungstechnik</p> <ul style="list-style-type: none"> • Digital-Analog-Wandler • Analog-Digital-Wandler <p>Schaltungen zur Realisierung der Schnittstelle zwischen Analog- und Digitaltechnik</p> <ul style="list-style-type: none"> • Funktionsprinzip von CMOS-Schaltungen • Einführung logischer Grundsaltungen in CMOS • Prinzipien mikroelektronischer Datenspeicher
6	Lernziele und Kompetenzen	<p>Die Studierenden:</p> <ul style="list-style-type: none"> • 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

		<ul style="list-style-type: none"> • erklären die Funktionsweise einfacher digitaler Schaltungen • entwickeln ein Verständnis 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: 6
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
10	Studien- und Prüfungsleistungen	<p>Klausur (90 Minuten)</p> <ul style="list-style-type: none"> • 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 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	<p>Die Studierenden:</p> <ul style="list-style-type: none"> • 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 Spezifikationsprachen (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-Entwicklungsprozess heraus • erläutern verschiedene Aspekte des Software-Projektmanagements
7	Voraussetzungen für die Teilnahme	Keine
8	Einpassung in Studienverlaufsplan	Semester: 6
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien 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 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	<p>Zahlreiche Realisierungen eingebetteter Systeme (z.B. Mobiltelefone, 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.</p> <p>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.</p> <p>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</p> <p>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.</p> <p>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.</p>

		<ol style="list-style-type: none"> 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	<p>Fachkompetenz - Wissen</p> <ul style="list-style-type: none"> • Die Studierenden erhalten Einblick in ein aktuelles Forschungsgebiet. <p>Fachkompetenz - Verstehen</p> <ul style="list-style-type: none"> • Die Studierenden verstehen Grundlagen des modernen Systementwurfs. • Die Studierenden erklären Implementierungsalternativen für digitale Hardware/Software-Systeme. <p>Fachkompetenz - Anwenden</p> <ul style="list-style-type: none"> • Die Studierenden wenden grundlegende Algorithmen an, zur Analyse und Optimierung von Hardware/Software-Systemen. <p>Competences: Professional competence - Knowledge</p> <ul style="list-style-type: none"> • Students gain insight into a current field of research. <p>Professional competence - Understanding</p> <ul style="list-style-type: none"> • Students understand the basics of modern system design. • Students explain implementation alternatives for digital hardware/software systems. <p>Professional competence - Application</p> <ul style="list-style-type: none"> • 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: 1
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<p data-bbox="611 306 1361 383">Empfohlene Bücher zur Begleitung und Vertiefung:</p> <ul data-bbox="651 461 1393 674" style="list-style-type: none"> <li data-bbox="651 461 1393 562">• Teich J., Haubelt C.: "Digitale Hardware/Software-Systeme: Synthese und Optimierung", Springer-Verlag, 2007, ISBN: 978-3-540-46822-6 <li data-bbox="651 573 1393 674">• Gajski, D. et al.: "Specification and Design of Embedded Systems", Prentice Hall, Englewood Cliffs, NJ, 1994, ISBN: 978-0131507319 <p data-bbox="611 752 1042 790">Weitere Informationen:</p> <p data-bbox="611 898 1393 965">https://www.cs12.tf.fau.de/lehre/lehveranstaltungen/vorlesungen/hardware-software-co-design</p>

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 Wolfgang Mehringer Anastasiya Zakreuskaya
5	Inhalt	<p>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.</p> <p>Die folgenden Themen werden im Modul behandelt:</p> <ul style="list-style-type: none"> • 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 <p>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:</p> <ul style="list-style-type: none"> • 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

		<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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. <p>Learning Objectives and Competences:</p> <ul style="list-style-type: none"> • 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	<p>Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007</p> <p>Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20222</p>
10	Studien- und Prüfungsleistungen	<p>Klausur</p> <p>Electronic exam (in presence), 90min</p>

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 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 Wolfgang Mehringer Anastasiya Zakreuskaya
5	Inhalt	<p>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.</p> <p>Die folgenden Themen werden im Modul behandelt:</p> <ul style="list-style-type: none"> • 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 <p>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:</p> <ul style="list-style-type: none"> • 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

		<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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. <p>Learning Objectives and Competences:</p> <ul style="list-style-type: none"> • 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: 6
9	Verwendbarkeit des Moduls	<p>Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007</p> <p>Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20222</p>
10	Studien- und Prüfungsleistungen	<p>Klausur</p> <p>Electronic exam (in presence), 90min</p>

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 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	<p>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, Merkmalsrepräsentationen 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.</p> <p>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.</p> <p>T</p>
6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <ul style="list-style-type: none"> • 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

		<ul style="list-style-type: none"> • 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 <p>The students</p> <ul style="list-style-type: none"> • 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	<p>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).</p> <p>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</p>

		thus creates the basis for more advanced modules (Pattern Recognition and Pattern Analysis).
8	Einpassung in Studienverlaufsplan	Semester: 4
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<ul style="list-style-type: none"> • 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 93107	Knowledge Discovery in Databases mit Übung Knowledge discovery in databases with tutorial	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. Richard Lenz	
5	Inhalt	<p>Theoretisches Wissen zu:</p> <ul style="list-style-type: none"> • Warum Data Mining? • Was ist Data Mining? • Eine multidimensionale Sicht auf Data Mining • Welche Arten von Daten können gewonnen werden? • Welche Arten von Mustern können gewonnen werden? • Welche Techniken werden benutzt? • Welche Anwendungen werden anvisiert? • Größere Probleme im Data Mining • Ein kurzer geschichtlicher Hintergrund zu Data Mining <p>Praktische Übungen zu:</p> <ul style="list-style-type: none"> • Einführung in Pandas & scikit-learn • Datenanalyse und -verarbeitung • Häufige Muster • Klassifikation • Clustering • Ausreißer 	
6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <ul style="list-style-type: none"> • 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; 	

		<ul style="list-style-type: none"> • legen dar, wie ein Entscheidungsbaum auf einem Trainingsdatensatz erzeugt wird; • stellen das Prinzip der Bayesschen 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.
7	Voraussetzungen für die Teilnahme	Empfohlen: Konzeptionelle Modellierung
8	Einpassung in Studienverlaufsplan	Semester: 4
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<p>Die Vorlesung basiert auf dem folgenden Buch:</p> <ul style="list-style-type: none"> • <ul style="list-style-type: none"> ◦ A. Géron, Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems, 2nd ed. O'Reilly 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

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	<p>Dieses Modul beschäftigt sich mit den Grundlagen der Künstlichen Intelligenz (KI), insbesondere formale Wissensrepräsentation, Heuristische Suche, Automatisches Planen und Schliessen unter Unsicherheit.</p> <p>---</p> <p>This module covers the foundations of Artificial Intelligence (AI), in particular symbolic techniques based on search and inference.</p>	
6	Lernziele und Kompetenzen	<ul style="list-style-type: none"> - 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 <p>*Inhalt*:</p> <ul style="list-style-type: none"> - 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. <p>---</p> <p>Technical, Learning, and Method Competencies</p> <ul style="list-style-type: none"> - 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). <p>Contents: Foundations of symbolic AI, in particular:</p>	

		<ul style="list-style-type: none"> - Agent Models as foundation of AI - Logic Programming in Prolog - Heuristic Search as a method 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	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<p>Die Vorlesung folgt weitgehend dem Buch</p> <p>Stuart Russell und Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition, 2009.</p> <p>Deutsche Ausgabe:</p> <p>Stuart Russell und Peter Norvig: Künstliche Intelligenz: Ein Moderner Ansatz. Pearson-Studium, 2004 (Übersetzung der 2. Auflage). ISBN: 978-3-8273-7089-1.</p>

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	<p>Fach- Lern- bzw. Methodenkompetenz</p> <ul style="list-style-type: none"> - 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. <p>Sozialkompetenz</p> <ul style="list-style-type: none"> - Die Studierenden arbeiten in Kleingruppen zusammen um kleine Projekte zu bewältigen. <p>Inhalte:</p> <ul style="list-style-type: none"> • Inferenz unter Unsicherheit • Bayessche Netzwerke • Rationale Entscheidungstheorie (MDPs and POMDPs) • Machinelles Lernen und Neuronale Netzwerke • Verarbeitung Natürlicher Sprache <p>---</p> <p>This course covers the foundations of Artificial Intelligence (AI), in particular reasoning under uncertainty, machine learning and (if there is time) natural language understanding.</p> <p>This course builds on the course Artificial Intelligence I from the preceding winter semester and continues it.</p> <p>Learning Goals and Competencies</p> <p>Technical, Learning, and Method Competencies</p> <ul style="list-style-type: none"> • 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. <p>Contents:</p> <ul style="list-style-type: none"> • Inference under Uncertainty

		<ul style="list-style-type: none"> • 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	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<p>Die Vorlesung folgt weitgehend dem Buch</p> <p>Stuart Russell und Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition, 2009.</p> <p>Deutsche Ausgabe:</p> <p>Stuart Russell und Peter Norvig: Künstliche Intelligenz: Ein Moderner Ansatz. Pearson-Studium, 2004 (Übersetzung der 2. Auflage).</p> <p>ISBN: 978-3-8273-7089-1.</p> <p>Literature</p> <p>The course follows the following textbook: Stuart Russell and Peter Norvig: Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition, 2009.</p>

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	<p>Fachkompetenz Wissen Die Studierenden beherrschen moderne, sehr expressive Formalismen zur Syntaktisch/Semantischen Analyse und Bedeutungsrepräsentation natürlicher Sprache. Sie können eingeschränkt neue Formalismen entwickeln.</p> <p>Anwenden Die Studierenden entwickeln Grammatiken und Bedeutungsrepräsentationen im Meta-Framework GF und setzen diese durch Interpretationsabbildungen in Verbindung.</p> <p>Analysieren Die Studierenden analysieren die innere Struktur natürlichsprachlicher Phrasen und Sätze. Sie wählen für eine zu repräsentierenden Phänomene geeignete Formalismen aus.</p> <p>Lern- bzw. Methodenkompetenz Die Studierenden erarbeiten selbständig formale syntaktische und semantische Repräsentationen für Natürliche Sprache</p> <p>Sozialkompetenz Die Studierenden arbeiten in Kleingruppen erfolgreich zusammen.</p> <p>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 Modellierung der Semantisch/proagmatischen Verarbeitung</p>	
7	Voraussetzungen für die Teilnahme	Keine	
8	Einpassung in Studienverlaufsplan	Semester: 1	

9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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 64640	Mathematik für Naturwissenschaften Mathematics for natural scientists	5 ECTS
2	Lehrveranstaltungen	Vorlesung mit Übung: Mathematik für Naturwissenschaftler (4 SWS) Übung: Übungen zur Mathematik für Naturwissenschaftler (2 SWS)	- -
3	Lehrende	Dr. Alexander Prechtel	

4	Modulverantwortliche/r	Dr. Alexander Prechtel
5	Inhalt	<ul style="list-style-type: none"> • Grundbegriffe der linearen Algebra und Analysis • Komplexe Zahlen • Lineare Abbildungen, Matrizen, Gauss-Algorithmus, Determinanten, Eigenwerte und Eigenvektoren, Diagonalisierung • Stetige und differenzierbaren Funktionen, Taylor-Reihen, Integralrechnung • Stabilitätsanalyse linearer Differentialgleichungssysteme <p>Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.</p>
6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <ul style="list-style-type: none"> • 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: 1
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien 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	Sämtliche Literatur mit Titel "Mathematik für Chemiker" oder "Ingenieursmathematik".
----	--------------------------	--

1	Modulbezeichnung 65760	Mathematische Modellbildung und Statistik für Naturwissenschaftler Mathematical modelling and statistics for natural scientists	5 ECTS
2	Lehrveranstaltungen	Vorlesung: Math. Modellbildung und Statistik für Naturwissenschaftler (3 SWS) Übung: R-Kurs zu "Math. Modellbildung und Statistik für Naturwissenschaftler" (1 SWS)	- -
3	Lehrende	Prof. Dr. Christophorus Richard	

4	Modulverantwortliche/r	Prof. Dr. Christophorus Richard	
5	Inhalt	<p>1. Grundbegriffe der Mathematik (Zahl, Vektor, Matrix, Zahlenfolge, Funktion, Ableitung)</p> <p>2. Funktionen (lineare und quadratische, e-Funktion, Logarithmusfunktionen)</p> <p>3. Beschreibende Statistik (ein- und zweidimensionale Stichproben, Lage-maße, Kovarianz, Korrelation, Zusammenhang zu linearer Regression)</p> <p>4. Verarbeitung von Sequenzdaten, Dotplots</p> <p>5. Wachstumsmodelle (lineares, exponentielles, logistisches und Variationen dazu, Allometrie, Modelle mit zeitlicher Verzögerung)</p> <p>6. Anpassung von Modellen an Daten (lineare Regression, logarithmische und doppellogarithmische Transformation von Daten)</p> <p>7. Modelle der chemischen Reaktionskinetik, incl. Michaelis-Menten-Modell</p> <p>8. Hardy-Weinberg Modell mit Variationen (Modellierung von Inzucht und Selektion)</p> <p>9. Grundbegriffe der Wahrscheinlichkeitstheorie: Binomialverteilung, Normalverteilung, Poissonverteilung und Zusammenhänge zwischen diesen Verteilungen</p> <p>10. Beurteilende Statistik: Testen (Binomialtest, verschiedene Chi²-Tests, t-Tests, Bedeutung der Freiheitsgrade")</p> <p>11. Beurteilende Statistik: Schätzen (Schätzer, Konfidenzintervall, Konfidenzband)</p> <p>12. Sequence-Alignment, Needleman-Wunsch Algorithmus</p> <p>13. Modelle für zwei Populationen: Räuber-Beute-Modell, Infektionsmodell</p> <p>Die Themen 1-6 und 9-12 werden in den Rechnerübungen durch praktische Aspekte ergänzt.</p>	
6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <ul style="list-style-type: none"> • können das Wechselspiel von mathematischer Modellierung und der Auswertung von Daten in biologisch relevanten Situationen erklären, • sind in der Lage, professionelle Statistiksoftware zur beschreibenden und schließenden Statistik für grundlegende Fragestellungen anzuwenden, 	

		<ul style="list-style-type: none"> • können die erlernten stochastische Konzepte und Modelle in konkreten Fragestellungen innerhalb eines vorgegebenen Zeitrahmens mit dem Rechner modellieren und erschöpfend analysieren; • sind in der Lage, verschiedene Modelle an Daten anzupassen.
7	Voraussetzungen für die Teilnahme	Keine
8	Einpassung in Studienverlaufsplan	Semester: 7
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20222
10	Studien- und Prüfungsleistungen	Studienleistung Klausur (50 Minuten) PL: Klausur 50 Min. SL: Praxisprüfung am Rechner (50 Min., E-Prüfung, unbenotet)
11	Berechnung der Modulnote	Studienleistung (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
16	Literaturhinweise	Schulwissen der Mathematik im Umfang von Abschnitt 2 bis 15 des Buches "Startwissen Mathematik und Statistik" von Harris, Taylor, Taylor (Spektrum Verlag 2007)

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	<p>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.</p> <p>Schlüsselwörter: Monaden, Funktionale Programmierung, Kategorientheorie, Haskell, Equational reasoning;</p> <p>Kursseite: https://www8.cs.fau.de/monad-based-programming/</p>
6	Lernziele und Kompetenzen	<h2>Fachkompetenz Wissen</h2> <p>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.</p> <h2>Anwenden</h2> <p>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.</p> <h2>Analysieren</h2> <p>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.</p>

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	<p>Als empfohlene Voraussetzung ist "Grundlagen der Logik in der Informatik" ausreichend.</p> <p>Alternativ sind allgemeine mathematische und logische Grundkenntnisse, die Sie auf jegliche Art und Weise erworben haben, ebenfalls geeignet.</p> <p>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):</p> <ul style="list-style-type: none"> - 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 $A \rightarrow B$ 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 $1^2 + \dots + n^2$? - Was ist die Definition einer teilweise geordneten Menge? - Was ist eine Äquivalenz?
8	Einpassung in Studienverlaufsplan	Semester: 1
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
10	Studien- und Prüfungsleistungen	<p>mündlich Prüfungsleistung, mündliche Prüfung, Dauer (in Minuten): 30, benotet, 7.5 ECTS</p> <p>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</p>

		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 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 zugrundeliegenden 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 Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007	
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		

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	<p>Fachkompetenz Wissen The students explain the basics of both programming semantics and proof assistants, in particular Coq.</p> <p>Verstehen The students prove theorems using a proof assistant.</p> <p>Anwenden The students transfer proofs into programs and programs into proofs.</p> <p>Analysieren The students examine behaviour of simple programs using formal semantics</p> <p>Evaluieren (Beurteilen) The students evaluate the role played by logic and type theory in scientific approach to programming.</p> <p>Erschaffen The students provide formal semantics to a simple programming language.</p>
7	Voraussetzungen für die Teilnahme	Keine
8	Einpassung in Studienverlaufsplan	Semester: 1
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<p>Online book "Software Foundations" http://www.cis.upenn.edu/~bcpierce/sf/</p> <p>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/</p> <p>Supplementary reading on the theory of programming: Types and Programming Languages</p> <p>Benjamin C. Pierce, The MIT Press</p> <p>Supplementary reading on Coq: Interactive Theorem Proving and Program Development</p> <p>Coq'Art: The Calculus of Inductive Constructions</p> <p>Series: Texts in Theoretical Computer Science. An EATCS Series</p> <p>Bertot, Yves, Casteran, Pierre</p>

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	<p>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.</p> <p>The lecture covers the following topics:</p> <ul style="list-style-type: none"> • 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	<p>Students are able to:</p> <ul style="list-style-type: none"> • 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: 6
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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: <ul style="list-style-type: none"> • 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 93230	Seminar Didaktik der Informatik Seminar: Teaching computer science	2,5 ECTS
2	Lehrveranstaltungen	Seminar: Seminar Didaktik der Informatik (2 SWS)	-
3	Lehrende	Prof. Dr. Marc-Pascal Berges	

4	Modulverantwortliche/r	Prof. Dr. Marc-Pascal Berges	
5	Inhalt	<ul style="list-style-type: none"> • 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	<p>Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, informatische Bildungsprozesse an ausgewählten Beispielen auf wissenschaftlicher Grundlage zu gestalten und zu reflektieren.</p> <p>Sie</p> <ul style="list-style-type: none"> • 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: 6	
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007	
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		

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	<p>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.</p> <p>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.</p> <p>Das Seminar bereitet auf die Teilnahme am Programmierwettbewerb der Universität Erlangen-Nürnberg Ende des Sommersemesters vor. Es besteht Teilnahmepflicht für diesen Wettbewerb.</p>	
6	Lernziele und Kompetenzen	<p>Nach der Teilnahme an diesem Seminar sind die Studierenden in der Lage:</p> <ul style="list-style-type: none"> • 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, • grundlegende 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	Semester: 6
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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 93163	Seminar Informatik in der Bildung Seminar: Computer science in education	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	Semester: 6
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	Deutsch
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	<p>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.:</p> <ul style="list-style-type: none"> • freie Konstruktionen, universelle Pfeile und adjungierte Funktoren • Äquivalenzfunktoren • Monaden: Eilenberg-Moore und Kleisli-Kategorien; Freie Monaden; Becks Satz • Kartesisch abgeschlossene Kategorien • Vollständige Halbordnungen (cpos), Einbettungen/ Projektionen, Limes-Kolimes-Koinzidenz, Lösung rekursive Domaingleichungen • Kan Erweiterungen • (symmetrische) monoidale Kategorien • Faktorisierungsstrukturen 	
6	Lernziele und Kompetenzen	<p>Fachkompetenz Wissen Die Studierenden geben Definitionen und Ergebnisse aus der Kategorientheorie und ihren Anwendungen in der Informatik wieder.</p> <p>Verstehen Die Studierenden erläutern grundlegende Methoden und Beweise der Kategorientheorie.</p> <p>Anwenden Die Studierenden wenden die Konzepte und Beweismethoden an, um einfache Problemstellungen kategoriell zu beschreiben und entsprechende Aussagen zu beweisen.</p> <p>Erschaffen Die Studierenden bereiten Resultate des Gebiets und ihre Herleitung eigenständig zur Präsentation auf.</p> <p>Lern- bzw. Methodenkompetenz Die Studierenden erarbeiten sich selbständig den Inhalt von Fachbüchern und wissenschaftlichen Veröffentlichungen und geben diesen in einem strukturierten Vortrag verständlich wieder.</p> <p>Sozialkompetenz Die Studierenden arbeiten in einer größeren Gruppe an komplexen Problemen und erläutern eigene Einsichten den Mitstudierenden in verständlicher Form.</p>	

7	Voraussetzungen für die Teilnahme	Grundkenntnisse der Kategorientheorie.
8	Einpassung in Studienverlaufsplan	keine Einpassung in Studienverlaufsplan hinterlegt!
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<ul style="list-style-type: none"> • 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.

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	<ul style="list-style-type: none"> • 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	<p>Fachkompetenz Wissen Die Studierenden geben elementare Definitionen und Fakten zu den behandelten Formalismen wieder.</p> <p>Verstehen Die Studierenden</p> <ul style="list-style-type: none"> • 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 <p>Anwenden Die Studierenden</p> <ul style="list-style-type: none"> • verfassen formale Spezifikationen sequentieller und nebenläufiger Programme • verifizieren einfache Programme gegenüber ihrer Spezifikation durch <p>Anwendung der relevanten Kalküle</p> <ul style="list-style-type: none"> • <ul style="list-style-type: none"> ◦ führen einfache Beweise über Programme mittels Induktion und Koinduktion <p>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</p> <p>Lern- bzw. Methodenkompetenz Die Studierenden beherrschen das grundsätzliche Konzept des Beweises als hauptsächliche Methode des Erkenntnisgewinns in der theoretischen Informatik.</p>	

		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	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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	<ul style="list-style-type: none"> • 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

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	Xingze Tian Prof. Dr.-Ing. Tobias Günther	

4	Modulverantwortliche/r	Prof. Dr.-Ing. Tobias Günther	
5	Inhalt	<p>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.</p> <p>The lecture covers the following topics:</p> <ul style="list-style-type: none"> • 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, aggregation, focus and context), • lies in visualization (human biases and rules of thumb), • applications (deep learning, medical visualization, optimization) <p>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.</p>	
6	Lernziele und Kompetenzen	<p>Students learn to:</p> <ul style="list-style-type: none"> • 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: 6
9	Verwendbarkeit des Moduls	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 2007
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.

Praktikum Informatik

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 1. Staatsprüfung für das Lehramt an Gymnasien 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 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	<p>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.</p> <p>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.</p>	
6	Lernziele und Kompetenzen	<p>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.</p>	
7	Voraussetzungen für die Teilnahme	Keine	
8	Einpassung in Studienverlaufsplan	Semester: 7	
9	Verwendbarkeit des Moduls	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Gymnasien 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	<p>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.</p> <p>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.</p> <p>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.</p> <p>Der schriftliche Bericht beinhaltet die Beschreibung und Dokumentation des Systems und der selbstgebauten IoT-Geraete sowie die Erkenntnisse der Sicherheitsanalyse.</p> <p>Zusammengefasster Inhalt:</p> <ul style="list-style-type: none"> • 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	<p>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.</p> <p>Selbstkompetenz Lernende koennen eigenstaendig, ohne Aufsicht und Anleitung, ein IoT-Geraet detailliert analysieren und zu festgelegten Zeitpunkten Ergebnisse vorweisen.</p>	

		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 Gymnasien 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	<p>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).</p> <p>Inhalte sind u.a.:</p> <ul style="list-style-type: none"> • 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 <p>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 wöchentlichen Meeting.</p> <p>In der initialen Lernphase bekommt ihr Vorlesungsmaterial und Übungen zur Verfügung gestellt. In der anschliessenden Projektphase wird an einer App gearbeitet. Aktuell ist dies die offizielle FAU-App, die neu entwickelt wird.</p>	
6	Lernziele und Kompetenzen	<p>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.</p> <p>Selbstkompetenz Studierende integrieren sich in Teams, planen das gemeinsame Vorgehen und reflektieren über das bisher Geleistete.</p> <p>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.</p>	
7	Voraussetzungen für die Teilnahme	Keine	
8	Einpassung in Studienverlaufsplan	Semester: 7	

9	Verwendbarkeit des Moduls	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Gymnasien 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 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	<p>Programmierungswettbewerbe 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.</p> <p>Neben fachlichem Verständnis ist die Teamstrategie entscheidend für den Erfolg der Gruppe. Die Wettbewerbe werden auf drei Ebenen abgehalten:</p> <ul style="list-style-type: none"> • 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. <p>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:</p> <ul style="list-style-type: none"> • 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.

		<p>Die Studierenden erhalten eine entsprechende Rückmeldung und es dürfen beliebig viele Lösungsversuche eingereicht werden.</p> <ul style="list-style-type: none"> • 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 Teilnahme am Trainingslager ist verpflichtend, um die Bewertung der Teamfähigkeit zu ermöglichen. • Einzelcontests (bis Mitte September): Nach dem Trainingslager 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. <p>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.</p>
6	<p>Lernziele und Kompetenzen</p>	<p>*A - Methodenkompetenz*</p> <p>Die Studierenden</p> <ul style="list-style-type: none"> • 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 <p>*B - Selbst- und Sozialkompetenz:*</p> <p>Die Studierenden</p> <ul style="list-style-type: none"> • 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

		<ul style="list-style-type: none"> 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 Gymnasien Informatik 20222
10	Studien- und Prüfungsleistungen	<p>Praktikumsleistung</p> <p>Das Praktikum setzt sich aus drei Komponenten zusammen:</p> <ul style="list-style-type: none"> 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 Teilnahme am Trainingslager ist verpflichtend, um die Bewertung der Teamfähigkeit zu ermöglichen. Einzelcontests (bis Mitte September): Nach dem Trainingslager 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	<p>This course teaches agile methods (Scrum and XP) and open source tools using a single semester-long project.</p> <p>Topics covered are:</p> <ul style="list-style-type: none"> • 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 <p>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:</p> <ul style="list-style-type: none"> • 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. <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>The course information will also tell you how the course will be held (online or in person).</p>
6	Lernziele und Kompetenzen	<ul style="list-style-type: none"> • Students learn about software products and software development in an industry context • Students learn about agile methods, in particular, Scrum and Extreme Programming

		<ul style="list-style-type: none"> 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 Gymnasien 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 93197	Praktikum: Entwicklung interaktiver eingebetteter Systeme Laboratory course: Development of cyber-physical embedded systems	10 ECTS
2	Lehrveranstaltungen	Praktikum: Entwicklung interaktiver eingebetteter Systeme (8 SWS)	10 ECTS
3	Lehrende	Joachim Falk	

4	Modulverantwortliche/r	Joachim Falk
5	Inhalt	<p>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.</p> <p>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.</p> <p>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.</p> <ul style="list-style-type: none"> • 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	<p>Fachkompetenz - Wissen</p> <ul style="list-style-type: none"> Die Studierenden legen Konzepte des modellbasierten Entwurfs eingebetteter Systeme dar. <p>Fachkompetenz - Verstehen</p> <ul style="list-style-type: none"> 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. <p>Fachkompetenz - Anwenden</p> <ul style="list-style-type: none"> 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. <p>Selbstkompetenz</p> <ul style="list-style-type: none"> Die Studierenden schätzen ihre individuellen Stärken ab, um eine geeignete Aufteilung innerhalb der Gruppe festzulegen. <p>Sozialkompetenz</p> <ul style="list-style-type: none"> 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 Gymnasien 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	<p>Weitere Informationen:</p> <p>https://www.cs12.tf.fau.de/lehre/lehveranstaltungen/praktika/entwicklung-interaktiver-eingebetteter-systeme/</p>

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	<p>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.</p> <p>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.</p>
6	Lernziele und Kompetenzen	<p>Fachkompetenz - Erschaffen</p> <ul style="list-style-type: none"> Die Studierenden erstellen Lösungsideen für die Projekte und implementieren diese in Java für Lego Mindstorms Roboter. <p>Selbstkompetenz</p> <ul style="list-style-type: none"> Die Studierenden schätzen ihre Stärken ab, um eine geeignete Arbeitsaufteilung innerhalb der Gruppe zu finden. <p>Sozialkompetenz</p> <ul style="list-style-type: none"> 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	Semester: 7
9	Verwendbarkeit des Moduls	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Gymnasien 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	<p>Weitere Informationen:</p> <p>https://www.cs12.tf.fau.de/lehre/lehveranstaltungen/praktika/lego-mindstorms-praktikum/</p>

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	<p>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.</p> <p>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.</p>	
6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <ul style="list-style-type: none"> • 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 <p>The students</p> <ul style="list-style-type: none"> • 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 Gymnasien 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 &	

		<p>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.</p>
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 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 Gymnasien 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 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: <ul style="list-style-type: none"> • 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 Gymnasien 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	

