Friedrich-Alexander-Universität
Erlangen-Nürnberg

Modulhandbuch

für den Studiengang

1. Staatsprüfung für das Lehramt an Gymnasien Informatik

(Prüfungsordnungsversion: 20242)

für das Wintersemester 2024/25

Inhaltsverzeichnis

Didaktik der Informatik 1 (93211) ... 3
Grundlagen der Programmierung (93104) ... 6
Didaktik der Informatik 2 (93224) ... 8
Sichere Systeme (93105) ... 10
Didaktik der Informatik 3 (93225) ... 12
Einführung in die Algorithmik (93106) ... 14
Theoretische Informatik für Wirtschaftsinformatik und Lehramt (93201) ... 16
Parallele und Funktionale Programmierung (93040) ... 18
Einführung in das Software Engineering (93097) ... 20
Einführung in Datenbanken (93008) ... 22
Praktikum Maschinenprogrammierung (93085) ... 24
Rechnerkommunikation (93150) ... 26
Grundlagen der Systemprogrammierung (93181) ... 28
Grundlagen des Maschinellen Lernens und der Künstlichen Intelligenz (93095) ... 30
Wahlpflichtbereich Informatik Gymnasium
- Mathematische Modellbildung und Statistik für Naturwissenschaftler (65760) ... 33
- Staatsexamensvorbereitung Lehramt Informatik (93088) ... 35
- Berechenbarkeit und Formale Sprachen (93010) ... 37
- Grundlagen der Logik in der Informatik (93072) ... 39
- Grundlagen der Rechnerarchitektur und -organisation (93080) ... 42
- Analyse und Design objektorientierter Softwaresysteme mit der Unified Modeling Language (UML) (510375) ... 44
- Human Computer Interaction (645618) ... 46
- Forensische Informatik (792501) ... 49
- Algorithmik kontinuierlicher Systeme (93000) ... 51
- Theorie der Programmierung (93121) ... 53
- Zweitfachspezifische Module (LAGY INF) ...
  - Mathematik für Naturwissenschaftler (64640) ... 55

Praktikum Informatik
- Grafik-Praktikum Game Programming (240715) ... 58
- Praktikum: Entwicklung interaktiver eingebetteter Systeme (93197) ... 59
- IoT Security (93199) ... 62
- NWERC Praktikum (93129) ... 64
- Praktikum Mustererkennung (93155) ... 68
- Softwareentwicklungspraktikum Lehramt (93162) ... 70
- Mobile Application Development and Security (93203) ... 72
- Praktikum Lego Mindstorms (278855) ... 74
- Praktikum angewandte Systemsoftwaretechnik (113845) ... 76
- Praktikum Enterprise Computing (594684) ... 77

1	**Modulbezeichnung** 93211	**Didaktik der Informatik 1** Teaching computer science I	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Einführung in die Fachdidaktik Informatik (2 SWS, WiSe 2024) Vorlesung mit Übung: Curriculare Themen der Fachdidaktik Informatik (LA GYM) (2 SWS, SoSe 2025) Vorlesung mit Übung: Curriculare Themen der Fachdidaktik Informatik (LA RS/LA MS/MEd BPT) (2 SWS, SoSe 2025)	2,5 ECTS 2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr. Marc-Pascal Berges	

4	**Modulverantwortliche/r**	Prof. Dr. Marc-Pascal Berges
5	**Inhalt**	• Grundlagen der informatikbezogenen Unterrichtsplanung und -gestaltung • Informatik und Informatikdidaktik im Wissenschaftskontext • Informatische Modellbildung • Programmieren im Informatikunterricht • Werkzeuge für den Informatikunterricht • Unterrichtsmethoden und -techniken • Aufgaben und Aufgabenkultur für einen kompetenzorientierten Informatikunterricht
6	**Lernziele und Kompetenzen**	Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, Informatikunterricht begründet zu planen, durchzuführen und zu reflektieren. Sie sind in der Lage, Unterrichtsinhalte motivierend, schülernah, verständlich und zielführend zu vermitteln. Sie können Elemente der Informatik in Alltagssituationen zur Motivation und als Modellierungsgrundlage heranziehen, Realsituationen informatisch modellieren, den Prozess des Modellierens schülerbezogen gestalten und Schülerinnen und Schüler beim Modellieren unterstützen. Sie • analysieren informatische Unterrichtsgegenstände fachdidaktisch und geben Unterrichtsziele outcomeorientiert an • charakterisieren die Wissenschaft Informatik und ihre Rolle im Bildungskontext (Computer Literacy, Great Principles of Computing, Computational Thinking) und geben eine eigene Definition für Informatik an • geben Ziele des Informatikunterrichts (gemäß Lehrplan Bayern) an und beschreiben beispielhaft Möglichkeiten zur Umsetzung dieser Ziele • geben zu Inhalten des Lehrplans konkrete durch die SuS zu erwerbende Kompetenzen an und gestalten entsprechenden Unterricht • beschreiben die "roten Fäden" in den Lehrplänen für Informatik in Bayern und berücksichtigen diese in der Gestaltung von Unterricht • erläutern den Informationszentrierten Ansatz und seinen Einfluss auf den bayerischen Lehrplan

		• ordnen Inhalte des Lehrplans dem Gesamtkonzept des Lehrplans zu • beschreiben Informatische Modellbildung, geben Beispiele und Darstellungsformen für Modellierungstechniken an und begründen die Relevanz informatischen Modellierens für die Schulinformatik • erläutern und illustrieren den Modellbegriff und Modellbildungsprozess aus Sicht der Informatik an selbst gewählten Beispielen • wenden Theorie und Begriffe informatischer Modellbildung in der Gestaltung und Bewertung von Unterrichtsszenarien an • ordnen Beispiele und Werkzeuge des Informatikunterrichts den Klassen von Modellen zu (EIS) • diskutieren Stellenwert, Rolle und Ziele des Programmierens in der informatischen Bildung und im informationszentrierten Ansatz • diskutieren den Stellenwert von Modellierung und Programmierung im Informatikunterricht ihrer Schulform • grenzen die Begriffe Modellieren, Programmieren und Codieren voneinander ab • begründen aus historischer und aktueller Perspektive den Einsatz von Methoden und Werkzeugen für die Vermittlung von Programmierkompetenz • diskutieren den Einsatz visueller und textueller Programmiersprachen • wenden Werkzeuge für den Informatikunterricht begründet in der Gestaltung von Unterricht an. • nennen Kriterien für Werkzeuge und wählen Werkzeuge für den Informatikunterricht begründet aus • begründen den Einsatz der Projektmethode im Informatikunterricht und 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**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 3;4
9	**Verwendbarkeit des Moduls**	Module Fachdidaktik Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Portfolio 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**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	2 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• 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** 93104	**Grundlagen der Programmierung** Foundations of programming	**5 ECTS**
2	Lehrveranstaltungen	Übung: Übungen zu Grundlagen der Programmierung - 05 (2 SWS) Übung: Übungen zu Grundlagen der Programmierung - 01 (2 SWS) Übung: Übungen zu Grundlagen der Programmierung - 03 (2 SWS) Übung: Übungen zu Grundlagen der Programmierung - 04 (2 SWS) Übung: Übungen zu Grundlagen der Programmierung - 02 (2 SWS) Vorlesung: Grundlagen der Programmierung (2 SWS)	- - - - - -
3	Lehrende	Mathias Harrer Prof. Dr. Tim Weyrich	

4	**Modulverantwortliche/r**	Dr.-Ing. Vanessa Klein Prof. Dr. Tim Weyrich
5	**Inhalt**	• Grundlegende Begriffe: Problem, Algorithmus, Programm, Syntax, Semantik, von Neumann Architektur • Imperative Programmkonstrukte: Variablen, Zahlen, Strings, Arrays, Kontrollstrukturen, Methoden • Grundlagen asymptotische Aufwandsanalyse: Einführung O-Notation und einfache Abschätzungen • Robustes Programmieren: Exceptions, Assert, Testen, Verifikation, Debugging • Objektorientierte Programmierung: Klassen, Objekte, Vererbung, Polymorphie, Module • Datenstrukturen: Parametrisierte Typen, abstrakte Datentypen, Listen, dynamische Arrays, binäre Suche, Suchbäume, Hashtabellen
6	**Lernziele und Kompetenzen**	*Wissen:* Die Studierenden ... • erlernen die Grundlagen und das Vokabular der Programmierung anhand der Programmiersprache Java *Verstehen:* Die Studierenden ... • können algorithmische Beschreibungen in natürlicher Sprache verstehen • können einfache Algorithmen im Code verstehen und analysieren • verstehen die grundlegende Behälterdatentypen und deren Eigenschaften (insbesondere Laufzeit- und Speicherplatzbedarf ihrer Operationen) *Anwenden:* Die Studierenden ... • implementieren einfache Algorithmen in Java unter Verwendung verschiedener Kontrollstrukturen

		• strukturieren Java-Code in Paketen, Klassen und Methoden und entwickeln wiederverwendbare Funktionen • können einfache Komplexitätsanalysen erstellen (O-Kalkül) • benutzen verschiedene Möglichkeiten zur Absicherung gegen Fehlersituationen und zur Fehlerrückmeldung (Rückgabewert, Ausnahmebehandlung) • wenden geeignete Testverfahren an • kennen die Konzepte der objektorientieren Programmierung und können diese einsetzen • setzen Verfahren und Werkzeuge zur systematischen Lokalisierung und Behebung von Programmfehlern an (Debugging) und verbessern ihre Lösungen auf diese Weise iterativ • verwenden generische Behälterdatentypen sachgerecht in eigenen Programmen • setzen Lambda-Ausdrücke effektiv ein
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 1
9	**Verwendbarkeit des Moduls**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
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** 93224	**Didaktik der Informatik 2** Teaching computer science II	**5 ECTS**
2	Lehrveranstaltungen	Praktikum: Praktikum zur Anwendung von Informatiksystemen aus fachdidaktischer Sicht (4 SWS) Seminar: Hauptseminar Didaktik der Informatik (2 SWS)	2,5 ECTS 2,5 ECTS
3	Lehrende	Claudia Neuner Prof. Dr. Marc-Pascal Berges	

4	**Modulverantwortliche/r**	Prof. Dr. Marc-Pascal Berges
5	**Inhalt**	• Theoretische Fundierung der Didaktik der Informatik • Voraussetzungen und Rahmenbedingungen für Informatikunterricht • Lern- und Kompetenzziele des Informatikunterrichts
6	**Lernziele und Kompetenzen**	Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, Informatikunterricht zu planen, durchzuführen, zu reflektieren und auf wissenschaftlicher Grundlage weiterzuentwickeln. Sie haben vertiefte fachbezogene Reflexionskompetenzen, informatikdidaktische Basis- und diagnostische Kompetenzen sowie informatikunterrichtsbezogene Handlungskompetenzen erworben. Sie sind in der Lage, Unterrichtsinhalte motivierend, schülernah, verständlich und zielführend zu vermitteln. Hierzu können sie entscheiden, welche Inhalte der Informatik für die Schule relevant sind und diese lerngruppenadäquat aufbereiten. Sie können Elemente der Informatik in Alltagssituationen zur Motivation und als Modellierungsgrundlage heranziehen und den Beitrag des Faches zur Allgemeinbildung beschreiben und transportieren. Sie können Realsituationen informatisch modellieren, den Prozess des Modellierens schülerbezogen gestalten und Schülerinnen und Schüler beim Modellieren unterstützen.
7	**Voraussetzungen für die Teilnahme**	Es wird eine vorherige erfolgreiche Teilnahme am Modul "Didaktik der Informatik 1" (93211) wird empfohlen
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Module Fachdidaktik Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Portfolio 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**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 60 h
14	**Dauer des Moduls**	1 Semester

15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Hubwieser, Peter. Didaktik der Informatik. Springer-Verlag, 2007. • Schubert, Sigrid, and Andreas Schwill. Didaktik der Informatik. Spektrum Akademischer Verlag, 2011.

1	**Modulbezeichnung** 93105	**Sichere Systeme** Secure Systems	**5 ECTS**
2	Lehrveranstaltungen	Übung: Sichere Systeme Übung 5 (2 SWS) Übung: Sichere Systeme Übung 9 (2 SWS) Übung: Sichere Systeme Übung 4 (2 SWS) Übung: Sichere Systeme Übung 8 (2 SWS) Übung: Sichere Systeme Übung 6 (2 SWS) Übung: Sichere Systeme Übung 3 (2 SWS) Übung: Sichere Systeme Übung 1 (2 SWS) Übung: Sichere Systeme Übung 2 (2 SWS) Übung: Sichere Systeme Übung 7 (2 SWS) Übung: Sichere Systeme Übung 10 (2 SWS) Übung: Sichere Systeme Übung 12 (2 SWS) Übung: Sichere Systeme Übung 11 (2 SWS) Vorlesung: Sichere Systeme (2 SWS)	2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS
3	Lehrende	Prof. Dr.-Ing. Felix Freiling Maximilian Eichhorn	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Felix Freiling
5	**Inhalt**	Die Vorlesung gibt einen einführenden Überblick über Konzepte und Methoden der IT-Sicherheit. Themen (unter anderem): • Angreifer und Schutzziele • Cyberkriminalität und Strafbarkeit • Ethik und Privatsphäre • grundlegende Muster von Unsicherheit in technischen Systemen • grundlegende Sicherheitsmechanismen • Techniken der Sicherheitsanalyse • ausgewählte Beispiele aus dem Bereich der Kryptographie und Internetsicherheit (Web-Security) In der Übung werden die Themen der Veranstaltung beispielhaft eingeübt. Themen (unter anderem): • Kryptanalyse und Angreifbarkeit kryptographischer Protokolle • Schutzziele und Strafbarkeit • Zertifikate und Public-Key-Infrastrukturen • Web-Security • anonyme Kommunikation • formale Sicherheitsanalyse • Sicherheitstesten
6	**Lernziele und Kompetenzen**	Die Teilnehmenden 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

		Schwächen in Internetprotokollen erkennen und benennen. Sie können außerdem erläutern, wie man diese Schwachstellen ausnutzt und welche technischen und organisatorischen Maßnahmen geeignet sind, diese Schwachstellen zu vermeiden. Die Studierenden lernen, die Wirksamkeit von IT-Sicherheitsmechanismen im gesellschaftlichen Kontext und in Kenntnis professioneller Strukturen der Cyberkriminalität aus technischen, ethischen und rechtlichen Perspektiven zu bewerten.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 1
9	**Verwendbarkeit des Moduls**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
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**	• 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** 93225	**Didaktik der Informatik 3** Teaching computer science 3	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Extracurriculare Themen der Didaktik der Informatik (2 SWS, SoSe 2025) Vorlesung: Praktikum Informatik in der Bildung (4 SWS, SoSe 2025)	2,5 ECTS 5 ECTS
3	Lehrende	Prof. Dr. Marc-Pascal Berges	

4	**Modulverantwortliche/r**	Prof. Dr. Marc-Pascal Berges
5	**Inhalt**	• Didaktik der Informatik als Wissenschaft • Gamebased Learning im Informatikunterricht • Informatikwettbewerbe • Digitalisierung • Lehr-/Lernlabore • Physical Computing
6	**Lernziele und Kompetenzen**	Nach Abschluss des Moduls sind die Studierenden in der Lage, fachdidaktische Problemstellungen auch außerhalb der Schulcurricula zu reflektieren. Sie sind in der Lage, extracurriculare Inhalte zu planen und durchzuführen und geeignete Anküpfungspunkte an die curriculare Lehre zu identifizieren. Sie • erkennen die Fachdidaktik Informatik als eigenständige Wissenschaft und können aktuelle Fragestellungen nachvollziehen • erstellen Unterrichtsentwürfe nach der Theorie des gamebased learning • konzipieren und erzeugen Unterrichtsmaterialien (z.B. auf Basis von physical computing) • kennen die gängisten Wettbewerbe für Informatik • beschreiben die Rolle der Informatikdidaktik im Rahmen der Digitalisierung • wenden die Methode des explorativen Lernens auf Unterrichtsmaterialien und im Rahmen von Lehr-/Lernlaboren an.
7	**Voraussetzungen für die Teilnahme**	Es wird dringend empfohlen, die Module "Didaktik der Informatik 1" (93211) und "Didaktik der Informatik 2" (93224) vorab erfolgreich abgeschlossen zu haben.
8	**Einpassung in Studienverlaufsplan**	Semester: 8;9
9	**Verwendbarkeit des Moduls**	Module Fachdidaktik Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Portfolio 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**	Portfolio (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 60 h
14	**Dauer des Moduls**	2 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Hubwieser, Peter. Didaktik der Informatik. Springer-Verlag, 2007. • Schubert, Sigrid, and Andreas Schwill. Didaktik der Informatik. Spektrum Akademischer Verlag, 2011.

1	**Modulbezeichnung** 93106	**Einführung in die Algorithmik** Introduction to algorithms	**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**	Carina Harrius Christian Riess
5	**Inhalt**	Die Vorlesung "Einfuhrung in die Algorithmik" gibt eine fundierte Einfuhrung in die Gebiete der Algorithmen und Datenstrukturen. Diese Einfuhrung umfasst grundlegende Designkonzepte von Algorithmen und deren formale Analyse. Folgende Themen werden behandelt: • Grundlagen • Design und Analyse von Algorithmen Korrektheit von Algorithmen • Wachstumsfunktionen • Rekurrenz • Probabilistische Algorithmen und deren Analyse • Grundlegende Algorithmen und Datenstrukturen und deren formale Analyse • Datenstrukturen Sortierverfahren Graphalgorithmen • Ausgewahlte Themen • Algorithmen in der Zahlentheorie String matching • Matrix Operationen
6	**Lernziele und Kompetenzen**	Die Studierenden erwerben eine grundlegende Einfuhrung in die Konzepte und Methoden aus dem Bereich der Algorithmen und Datenstrukturen. Die Teilnehmer kennen grundlegende Techniken und Prinzipien zum Design von Algorithmen und Datenstrukturen. Die Studierenden kennen grundlegende Algorithmen im Bereich der Sortierung, der Graphentheorie und der Zahlentheorie. Des Weiteren kennen die Studierenden die notwendigen Datenstrukturen und verstehen deren Vor- und Nachteile in Bezug auf deren Effizienz und Komplexitat. Die Studierenden konnen die unterschiedlichen Designparadigmen von Datenstrukturen und Algorithmen auf neue Probleme anwenden und deren Korrektheit formal analysieren. Aus der Analyse konnen die Studierenden Algorithmen bewerten und vergleichen.
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 20242

10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (90 Minuten) Zu jedem der vier Themenblöcke der Vorlesung gibt es ein bewertetes Übungsblatt. Diese Übungsblätter können in Gruppen von bis zu vier Teilnehmern bearbeitet werden. Zu erreichen sind mindestens 50% der Punkte der Übungsblätter.
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%) Die Modulnote wird durch die Abschlussklausur bestimmt.
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**	• Introduction to Algorithms, Thomas H. Cormen , Charles E. Leiserson, Ronald L. Rivest, Clifford Stein Weitere Literatur wird in der Vorlesung bekannt gegeben.

1	**Modulbezeichnung** 93201	**Theoretische Informatik für Wirtschaftsinformatik und Lehramt** Theoretical computer science for information systems and teaching degree students	**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**	Grundlegende Begriffe und Kernergebnisse der Automatentheorie, Berechenbarkeitstheorie und Komplexitätstheorie werden überblickhaft behandelt: • endliche Automaten und reguläre Grammatiken und Sprachen • Kellerautomaten, kontextfreue Grammatiken und Sprachen • Turingmaschinen und berechenbare Funktionen • Primitiv rekursive und mü-rekursive Funktionen • LOOP- und WHILE-Berechenbarkeit • Entscheidbare Sprachen und Unentscheidbarkeit • Chomsky-Hierarchie • Komplexitätsklassen P und NP • NP-Vollständigkeit
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben elementare Definitionen und Fakten zu formalen Sprachen und entsprechenden Maschinenmodellen und Grammatiken wieder. Verstehen Die Studierenden • erklären grundlegende Konzepte der Begriffe der Automaten- und Berechenbarkeits- und Komplexitätstheorie. • beschreiben Beispiele dieser Konzepte. • erläutern grundlegende Konstruktionen, Algorithmen und wesentliche Resultate und entsprechende Beweise (z.B. Unentscheibarkeit des Halteproblems). Anwenden Die Studierenden • führen Konstruktionen auf vorgelegten Maschinen und Grammatiken und Sprachen durch (z.B. Automatenminiierung, Potenzmengen-Konstruktion, Chomsky-Normierung, CYK-Algorithmus). • wenden grundlegende Beweisverfahren der theoretischen Informatik an (z.B. Induktionsbeweise, Pumping-Lemma, Reduktionen). Analysieren Die Studierenden

		• analysieren formale Sprachen und ermitteln ihre Zugehörigkeit zu den Klassen der Chomsky-Hierarchie. • untersuchen die Entscheidbarkeit von vorgelegten formalen Sprachen. • analysieren die Komplexität eine Entscheidungsproblem und klassifizieren es als Problem in P, NP bzw. NP-vollständig. Lern- bzw. Methodenkompetenz Die Studierenden • beherrschen das grundsätzliche Konzept des Beweises als hauptsächliche Methode des Erkenntnisgewinns in der theoretischen Informatik. Sie überblicken abstrakte Begriffsarchitekturen. • vollziehen mathematische Argumentationen nach, erklären diese, führen diese selbst und legen sie schriftlich nieder. Sozialkompetenz Die Studierenden lösen Probleme in kollaborativer Gruppenarbeit und präsentieren erarbeitete Lösungen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 2
9	**Verwendbarkeit des Moduls**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 56 h Eigenstudium: 94 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• U. Schöning: Theoretische Informatik - kurz gefasst, 5. Aufl., Spektrum 2008. • J.E. Hopcroft, R. Motwani und J.D. Ullman: Introduction to Automata Theory, Languages, and Computation, 2. Aufl., Addison Wesley, 2001.

1	**Modulbezeichnung** 93040	**Parallele und Funktionale Programmierung** Parallel and functional programming	**5 ECTS**
2	Lehrveranstaltungen	Übung: PFP-T06 (2 SWS) Übung: PFP-T01 (2 SWS) Übung: PFP-T02 (2 SWS) Übung: PFP-R07 (2 SWS) Übung: PFP-T07 (2 SWS) Übung: PFP-R01 (2 SWS) Übung: PFP-R04 (2 SWS) Übung: PFP-T04 (2 SWS) Übung: PFP-T03 (2 SWS) Übung: PFP-R02 (2 SWS) Übung: PFP-R06 (2 SWS) Übung: PFP-T05 (2 SWS) Vorlesung: Parallele und Funktionale Programmierung (2 SWS)	2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS 2,5 ECTS
3	Lehrende	Julian Brandner Prof. Dr. Michael Philippsen David Schwarzbeck Dr.-Ing. Norbert Oster	

4	**Modulverantwortliche/r**	Dr.-Ing. Norbert Oster Prof. Dr. Michael Philippsen
5	**Inhalt**	• Grundlagen der funktionale Programmierung • Grundlagen der parallelen Programmierung • Datenstrukturen • Objektorientierung • Scala-Kenntnisse • Erweiterte JAVA-Kenntnisse • Aufwandsabschätzungen • Grundlegende Algorithmen
6	**Lernziele und Kompetenzen**	Die Studierenden • erlernen die Grundlagen der funktionalen Programmierung anhand der Programmiersprache Scala • verstehen paralleles Programmieren mit Java • kennen fundamentale Datenstrukturen und Algorithmen • können funktionale und parallele Algorithmen entwickeln und analysieren
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 3

9	**Verwendbarkeit des Moduls**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
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** 93097	**Einführung in das Software Engineering** Introduction to software engineering	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung: Introduction to Software Engineering Übung: Introduction to Software Engineering Exercises	- -
3	Lehrende	Sally Zeitler Prof. Dr.-Ing. Andreas Maier	

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

8	**Einpassung in Studienverlaufsplan**	Semester: 3
9	**Verwendbarkeit des Moduls**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Software Engineering, Ian Sommerville, 10. Auflage, 2016 • Software-Engineering Kompakt, Anja Metzner, 2020 • Handbook of Software Engineering, Sungdeok Cha, Richard N. Taylor, Kyochul Kang (Hrsg.), 2019

1	**Modulbezeichnung** 93008	**Einführung in Datenbanken** Introduction to databases	**7,5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Richard Lenz
5	**Inhalt**	Ziel des Moduls ist die Vermittlung von Kenntnissen zur systematischen und bedarfsorientierten Erstellung konzeptioneller Datenbankschemata sowie die relationale Datenbanksprache SQL. Darüber hinaus werden Grundkenntnisse zur Funktionsweise und zur Implementierung von Datenbankmanagementsystemen vermittelt, im Einzelnen: • Grundbegriffe von Datenbanken • Entity-Relationship Modell und erweitertes E/R-Modell • UML Klassendiagramme • Das Relationale Datenmodell • Systematische Abbildung von ER-Diagrammen auf Relationale Datenbankschemata • Normalisierung • Relationale Algebra • SQL • Multidimensionale Modellierung und Data Warehousing • Schichtenmodell zur Implementierung von Datenbanksystemen • Pufferverwaltung • Indexstrukturen (B-Bäume, B+-Bäume) • Anfrageverarbeitung • Transaktionen • Synchronisation • Recovery • Andere Datenmodelle, No-SQL Systeme • Ontologien, Semantic Web, RDF, SPARQL
6	**Lernziele und Kompetenzen**	Die Studierenden • Können die zentralen Begriffe aus der Datenbankfachliteratur definieren • Erstellen ER-Diagramme und erweiterte ER Diagramme • Können ER-Diagramme systematisch in geeignete relationale Datenbankschemata überführen • Definieren die Normalformen 1NF, 2NF, 3NF, BCNF und 4NF • Können ein nicht normalisiertes Relationenschema in 3NF überführen • Erstellen Anfragen auf der Basis der Relationalen Algebra • Erstellen Datenbankschemata mit Hilfe der SQL DDL • Erstellen Datenbankanfragen mit SQL • Erstellen multidimensionale ER-Diagramme und bilden diese auf Star- oder Snowflake-Schemata ab

		• Erklären die Funktionsweise von Datenbankpuffern • Erklären die Funktionsweise von Indexstrukturen • Erklären die Grundlagen der Anfrageoptimierung • Erläutern und bewerten die Funktionsweise verschiedener Join-Algorithmen • Erklären die ACID Eigenschaften von Transaktionen • Erklären die Funktionsweise des Zwei-Phasen-Freigabe-Protokolls • Erläutern die Funktionsweise des Zwei-Phasen-Sperr-Protokolls • Vergleichen die verschiedenen Klassen von Wiederherstellungs-Algorithmen • Erläutern die grundlegende Funktionsweise der Protokoll-basierten Wiederherstellung • Beschreiben und vergleichen verschiedene Datenmodelle
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 20242
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**	

1	**Modulbezeichnung** 93085	**Praktikum Maschinenprogrammierung** Machine code lab	**5 ECTS**
2	Lehrveranstaltungen	Praktikum: Praktikum Maschinenprogrammierung (4 SWS)	5 ECTS
3	Lehrende	Stefanie Senft Annabel Lindner	

4	**Modulverantwortliche/r**	Prof. Dr. Marc-Pascal Berges
5	**Inhalt**	Aufbau und Prinzip von Rechnern, Daten und ihre Codierung, Boolesche Algebra und Schaltalgebra, Schaltnetze (Symbole, Darstellung), Optimierung von Schaltnetzen (Minimierung Boolescher Funktionen), Realisierungsformen von Schaltnetzen (ROM, PLA, FPGA), Automaten und Schaltwerke (Moore/Mealy, Zustandscodierung und -minimierung), Flipflops, Register, Zähler, Speicher (RAM, ROM), Taktung und Synchronisation, Realisierungsformen von Schaltwerken, Realisierung der Grundrechenarten Addition/Subtraktion, Multiplikation und Division, Gleitkommazahlen (Darstellung, Fehler, Rundung, Standards, Einheiten), Steuerwerksentwurf, Spezialeinheiten und Co-Prozessoren, Mikrocontroller; vorlesungsbegleitende Einführung und Beschreibung der Schaltungen mit VHDL.
6	**Lernziele und Kompetenzen**	• 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. • Die Studierenden verstehen, dass Hardware heutzutage mit Software am Rechner entwickelt und simuliert wird. • Die Studierenden verstehen den Schaltungsentwurf mittels einer Beschreibungssprache (VHDL). • Die Studierenden erarbeiten und diskutieren verschiedene Lösungswege für die Datencodierung sowie den Entwurf und die Optimierung von digitalen Hardwareschaltungen. • Die Studierenden erlernen die Fähigkeit, digitale Schaltungen und Systeme eigenständig zu konzipieren und zu implementieren. • Die Studierenden reflektieren den Umgang mit schulgeeigneten Mikrocontrollern
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 20242
10	**Studien- und Prüfungsleistungen**	schriftlich/mündlich

		Die Prüfungsleistung besteht aus 8-10 praktischen Aufgaben sowie 3-4 schriftlichen Testaten
11	**Berechnung der Modulnote**	schriftlich/mündlich (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93150	**Rechnerkommunikation** Computer communications	**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. Reinhard German
5	**Inhalt**	Die Vorlesung vermittelt die Grundlagen der Rechnerkommunikation und durchläuft von oben nach unten die Schichten des Internets: • Anwendungsschicht • Transportschicht • Netzwerkschicht • Sicherungsschicht • Physikalische Schicht Sicherheit wird als übergreifender Aspekt behandelt. An verschiedenen Stellen werden analytische Modelle eingesetzt, um Wege für eine quantitative Auslegung von Kommunikationsnetzen aufzuzeigen. Die Übung beinhaltet praktische und theoretische Aufgaben zum Verständnis der einzelnen Schichten.
6	**Lernziele und Kompetenzen**	Die Studierenden erwerben • Kenntnisse über zentrale Mechanismen, Protokolle und Architekturen der Rechnerkommunikation (Topologie, Schicht, Adressierung, Wegsuche, Weiterleitung, Flusskontrolle, Überlastkontrolle, Fehlersicherung, Medienzugriff, Bitübertragung) am Beispiel des Internets und mit Ausblicken auf andere Netztechnologien • Kenntnisse über Sicherheit, Leistung und Zuverlässigkeit bei der Rechnerkommunikation • praktische Erfahrung in der Benutzung und Programmierung von Rechnernetzen
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4;6
9	**Verwendbarkeit des Moduls**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Übungsleistung Klausur (90 Minuten) Hausaufgaben zu Rechnerkommunikation (Übungsleistung): • Studienleistung, Übungsleistung, unbenotet, 2.5 ECTS • weitere Erläuterungen: Bearbeitung (zwei)wöchentlicher Aufgabenblätter in Gruppenarbeit. Für den unbenoteten Übungsschein sind 60% der Punkte je Aufgabenblatt zu erreichen Rechnerkommunikation (Klausur):

		• Prüfungsleistung, Klausur, Dauer (in Minuten): 90, benotet, 2.5 ECTS • Anteil an der Berechnung der Modulnote: 100.0 %
11	**Berechnung der Modulnote**	Übungsleistung (0%) Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Lehrbuch: Kurose, Ross. Computer Networking. 8th Ed., Pearson, 2021.

1	**Modulbezeichnung** 93181	**Grundlagen der Systemprogrammierung** Foundations of system programming	**5 ECTS**
2	Lehrveranstaltungen	Übung: SP-RÜ R11 (2 SWS) Übung: SP-RÜ Do-14, Ferdinand S. & Stefan S. (2 SWS) Übung: SP-RÜ Do-16, Johannes K. & Hannes S. (2 SWS) Übung: SP-RÜ R14 (2 SWS) Übung: SP-RÜ Fr-12, Christian H. (2 SWS) Übung: SP-RÜ R13, Kevin Kollenda (2 SWS) Übung: SP-RÜ Mo-16 (2 SWS) Übung: SP-RÜ Fr-14, Frederik Z. (2 SWS) Übung: SP-RÜ Fr-16, Lukas B. (2 SWS) Übung: SP-RÜ R01 (2 SWS) Übung: SP-RÜ Mi-16, Christian H. & Maximilian R. (2 SWS) Übung: SP-RÜ Mi-14 (2 SWS) Übung: SP-RÜ Di-12, Julian Z. & Dana E. (2 SWS) Übung: SP-RÜ R15 (2 SWS)	- - - - - - - - - - - - - -
3	Lehrende	Maximilian Ott Thomas Preisner Luis Gerhorst	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Wolfgang Schröder-Preikschat
5	**Inhalt**	• Grundlagen von Betriebssystemen (Adressräume, Speicher, Dateien, Prozesse, Koordinationsmittel; Betriebsarten, Einplanung, Einlastung, Virtualisierung, Nebenläufigkeit, Koordination/Synchronisation) • Abstraktionen/Funktionen UNIX-ähnlicher Betriebssysteme • Programmierung von Systemsoftware • C, Make, UNIX-Shell (Solaris, Linux, MacOS X)
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben Kenntnisse über Grundlagen von Betriebssystemen • verstehen Zusammenhänge, die die Ausführungen von Programmen in vielschichtig organisierten Rechensystemen ermöglichen • erlernen die Programmiersprache C • entwickeln Systemprogramme auf Basis der Systemaufrufschnittstelle UNIX-ähnlicher Betriebssysteme
7	**Voraussetzungen für die Teilnahme**	Keine

8	**Einpassung in Studienverlaufsplan**	Semester: 6
9	**Verwendbarkeit des Moduls**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Klausur mit MultipleChoice (90 Minuten)
11	**Berechnung der Modulnote**	Klausur mit MultipleChoice (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 60 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Lehrbuch: Betriebssysteme Grundlagen, Entwurf, Implementierung, Wolfgang Schröder-Preikschat, 2008

1	**Modulbezeichnung** 93095	**Grundlagen des Maschinellen Lernens und der Künstlichen Intelligenz** Fundamentals of machine learning and artificial intelligence	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Grundlagen des Maschinellen Lernens und der Künstlichen Intelligenz	-
3	Lehrende	Michaela Müller-Unterweger	

4	**Modulverantwortliche/r**	Prof. Dr. Marc-Pascal Berges
5	**Inhalt**	• Begriff der Intelligenz • Heuristiken und Suchstrategien • Entscheidungsbaum und k-nächster Nachbarn-Algorithmus (knN) • supervised learning • Beispiele und Anwendungen zum Entscheidungsbaum und zum knN • Orange oder Python-basierte Frameworks • PROLOG als regelbasierte Programmiersprache • Regelbasierte KI auf Basis von logischen Ausdrücken (Konjunktive Normalform und disjunktive Normalform) • reinforcement learning • Perzeptron als einzelnes Neuron zur Klassifikation anhand verschiedener Beispiele • Neuronale Netz als Basis für maschinelles Lernen • Implementieren eines Perzeptrons in Java • unsupervised learning • Back- / Forwardpropagation • Etische und gesellschaftliche Fragen zum Einsatz neuronaler Netze und künstlicher Intelligenz • Bilderkennung mit neuronalen Netzen in Orange und Python-basiert
6	**Lernziele und Kompetenzen**	Die Lernenden sind in der Lage, die grundlegenden Funktionsweisen maschinellen Lernens zu erklären. Sie sind in der Lage, Algorithmen aus dem Bereich des maschinellen Lernens selbstständig zu implementieren. Die Lernenden sind in der Lage wissens- und datenbasierte KI-System zu erläutern. Sie können zu ethischen und gesellschaftlichen Fragenstellungen mit Bezug zur künstlichen Intelligenz stellungnehmen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 8
9	**Verwendbarkeit des Moduls**	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) Prüfungsleistung ist eine Klausur im Umfang von 90 Minuten.

11	**Berechnung der Modulnote**	Klausur (100%) 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**	

Wahlpflichtbereich Informatik Gymnasium

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	apl. Prof. Dr. Christophorus Richard	

4	**Modulverantwortliche/r**	apl. Prof. Dr. Christophorus Richard
5	**Inhalt**	1. Grundbegriffe der Mathematik (Zahl, Vektor, Matrix, Zahlenfolge, Funktion, Ableitung) 2. Funktionen (lineare und quadratische, e-Funktion, Logarithmusfunktionen) 3. Beschreibende Statistik (ein- und zweidimensionale Stichproben, Lage-maße, Kovarianz, Korrelation, Zusammenhang zu linearer Regression) 4. Verarbeitung von Sequenzdaten, Dotplots 5. Wachstumsmodelle (lineares, exponentielles, logistisches und Variationen dazu, Allometrie, Modelle mit zeitlicher Verzögerung) 6. Anpassung von Modellen an Daten (lineare Regression, logarithmische und doppeltlogarithmische Transformation von Daten) 7. Modelle der chemischen Reaktionskinetik, incl. Michaelis-Menten-Modell 8. Hardy-Weinberg Modell mit Variationen (Modellierung von Inzucht und Selektion) 9. Grundbegriffe der Wahrscheinlichkeitstheorie: Binomialverteilung, Normalverteilung, Poissonverteilung und Zusammenhänge zwischen diesen Verteilungen 10. Beurteilende Statistik: Testen (Binomialtest, verschiedene Chi2-Tests, t-Tests, Bedeutung der Freiheitsgrade") 11. Beurteilende Statistik: Schätzen (Schätzer, Konfidenzintervall, Konfidenzband) 12. Sequence-Alignment, Needleman-Wunsch Algorithmus 13. Modelle für zwei Populationen: Räuber-Beute-Modell, Infektionsmodell Die Themen 1-6 und 9-12 werden in den Rechnerübungen durch praktische Aspekte ergänzt.
6	**Lernziele und Kompetenzen**	Die Studierenden • 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,

		• 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 20242
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** 93088	**Staatsexamensvorbereitung Lehramt Informatik** State Examination preparation for teaching degrees with computer science	**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 Algorithmik 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 20242
10	**Studien- und Prüfungsleistungen**	Variabel 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**	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** 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 Linus Gnan Franz Schlicht Marko Griesser-Aleksic Dominik Pysch Johanna Zaviska	

4	**Modulverantwortliche/r**	Prof. Dr. Rolf Wanka
5	**Inhalt**	• Registermaschinen und Turingmaschinen als Modelle des Berechenbaren, die Church-Turing-These und unentscheidbare Probleme • NP-Vollständigkeit und das P-NP-Problem • Endliche Automaten • Grammatiken und die Chomsky-Hierarchie • Kontextfreie Grammatiken und Kontextfreie Sprachen • Kellerautomaten
6	**Lernziele und Kompetenzen**	Die Studierenden • erwerben fundierte Kenntnisse über die Grenzen der Berechenbaren, insbesondere lernen sie, wie man beweist, dass bestimmte Aufgaben unlösbar sind bzw. dass sie vermutlich nicht schnell gelöst werden können, und wenden diese Kenntnisse an; • lernen die wesentlichen Techniken kennen, mit denen man Programmiersprachen beschreiben und syntaktisch korrekte Programme erkennen kann, und wenden diese auf Beispiele an; • erwerben fundierte Kenntnisse in den Beweis- und Analyse-Methoden der algorithmisch orientierten Theoretischen Informatik und wenden diese an.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 7
9	**Verwendbarkeit des Moduls**	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) Übungsleistung Zum Erreichen der Übungsleistung müssen die wöchentlichen bepunkteten Übungsaufgaben bearbeitet und abgegeben werden. Zum

		Ende der Vorlesungszeit müssen mindestens 50% der Punkte erreicht sein.
11	**Berechnung der Modulnote**	Klausur (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• I. Wegener. Theoretische Informatik. • J. Hopcroft, J. Ullman. Introduction to Automata Theory, Languages and Computation. • U. Schöning. Theoretische Informatik - kurz gefasst.

1	**Modulbezeichnung** 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 Zisis Erkelentzis Max Ole Elliger Nova Ruff Tom Weißhuhn David Wegmann Silas Kuder Jonas Hemkendreis Prof. Dr. Lutz Schröder	

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

		geben Definitionen zur Syntax und Semantik der verwendeten Logiken wieder beschreiben grundlegende Deduktionsalgorithmen geben Regeln der verwendeten formalen Deduktionssysteme wieder Verstehen Die Studierenden erläutern das Verhältnis zwischen Syntax, Semantik und Beweistheorie der verwendeten Logiken erklären die Funktionsprinzipien grundlegender Deduktionsalgorithmen erläutern die Funktionsweise automatischer Beweiser erläutern grundlegende Resultate der Metatheorie der verwendeten Logiken und deren Bedeutung Anwenden Die Studierenden wenden Deduktionsalgorithmen auf konkrete Deduktionsprobleme an formalisieren Anwendungsprobleme in logischer Form und verwenden automatische Beweiser zur Erledigung entstehender Beweisziele führen einfache formale Beweise manuell Analysieren Die Studierenden führen einfache metatheoretische Beweise, inbesondere durch syntaktische Induktion Lern- bzw. Methodenkompetenz Die Studierenden beherrschen das grundsätzliche Konzept des Beweises als hauptsächliche Methode des Erkenntnisgewinns in der theoretischen Informatik. Sie überblicken abstrakte Begriffsarchitekturen. Sozialkompetenz Die Studierenden lösen abstrakte Probleme in Gruppenarbeit.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 7
9	**Verwendbarkeit des Moduls**	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) Es werden wöchentlich Übungsblätter ausgegeben. Die Lösungen können abgegeben werden und werden in diesem Fall bewertet. Auf Basis des Ergebnisses dieser Bewertungen können bis zu 15% Bonuspunkte erworben werden, die zu dem Ergebnis einer bestandenen Klausur hinzugerechnet werden.
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	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

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

7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 6
9	**Verwendbarkeit des Moduls**	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Hennessy/Patterson: Computer Architecture - A quantitative approach, 4.Auflage, 2006, MorganKaufmann. Patterson/Hennessy: Computer Organization & Design, 4.Auflage, 2008, MorganKaufmann. Stallings, Computer Organization & Architecture, 8.Auflage, 2009, Prentice Hall. Märtin, Rechnerarchitekturen, 2001, Fachbuchverlag Leipzig.

1	**Modulbezeichnung** 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	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. Detlef Kips
5	**Inhalt**	Die sogenannte "Unified Modeling Language" (UML) ist der seit Jahren weltweit akzeptierte Notationsstandard für die Modellierung komplexer Softwaresysteme. Mit einem reichhaltigen Repertoire an graphischen und textuellen Ausdrucksmöglichkeiten bietet die UML ihren Anwendern die Möglichkeit, die Anforderungen an die Zielsoftware, ihre statischen bzw. dynamischen Systemeigenschaften sowie die gewählte Softwarearchitektur halbformal zu spezifizieren, im Team darüber zu kommunizieren und große Teile des Programmcodes aus den spezifizierten Systemmodellen zu generieren. Ziel dieser Vorlesung ist es, die Studierenden mit Syntax und Semantik der UML vertraut zu machen und zu demonstrieren, wie die UML im Rahmen eines "typischen" Softwareentwicklungsprozesses angewendet werden kann. Zu diesem Zweck werden die verschiedenen Diagrammtypen und Notationselemente der UML schrittweise eingeführt und anhand eines durchgängigen Anwendungsbeispiels im Rahmen eines konkreten Vorgehensmodells über alle Entwicklungsphasen hinweg eingesetzt.
6	**Lernziele und Kompetenzen**	Im Rahmen dieser Veranstaltung sollen die Studierenden insbesondere die Kompetenz erwerben, - die syntaktische Struktur und die Semantik vorgegebener UML-Modelle bzw. Modellausschnitte zu analysieren und zu erläutern - verschiedene Sprachelemente der UML (und ggf. deren Kombination) im Hinblick auf ihre Eignung zur Abbildung charakteristischer Modellierungsprobleme im Rahmen eines Softwareentwicklungsprozesses zu bewerten, auszuwählen und anzuwenden - die Struktur und Systematik des UML-Metamodells zu erläutern und die UML mit geeigneten Metamodellierungskonzepten auf spezifische Anwendungskontexte anzupassen - zu einer gegebenen Anforderungsdefinition im Rahmen einer systematischen Analyse- und Entwurfsmethodik ein integriertes UML-Systemmodell zu erstellen - den Funktionsumfang eines UML-basierten Modellierungswerkzeugs zu bewerten, ein geeignetes Werkzeug auszuwählen und sicher anzuwenden.

7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 6
9	**Verwendbarkeit des Moduls**	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	mündlich mündliche Einzelprüfung; Dauer (in Minuten): 30; benotet; 5 ECTS (Vorlesung + Übung)
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Rumbaugh, J.; Booch, G.; Jacobson, I.: The Unified Modeling Language Reference Manual, Addison-Wesley, 2004 • Hitz, M.; Kappel, G.; Kapsammer, E.; Retschitzegger, W.: UML @ work , 3., aktualisierte und überarbeitete Auflage, dpunkt-Verlag, 2005 • Winter, M.: Methodische objektorientierte Softwareentwicklung, dpunkt-Verlag, 2005 • Störrle, H.: UML 2 erfolgreich einsetzen, Addison-Wesley, 2007 • Rumpe, B.: Modellierung mit UML: Sprache, Konzepte und Methodik, Springer-Verlag, 2. Auflage, 2011 • Seidl, M., Brandsteidl, M., Huemer, C., Kappek, G.: UML@classroom - Eine Einführung in die objektorientierte Modellierung, dpunkt-Verlag, 2012 • Rupp, C.; Queins, S., et al. UML 2 glasklar: Praxiswissen für die UML-Modellierung, Carl Hanser Verlag, 2012 Die einschlägige Originalliteratur zur UML findet man auf der Website der Object Management Group (http://www.omg.org/spec/UML).

1	**Modulbezeichnung** 645618	**Human Computer Interaction** Human computer interaction	**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. Björn Eskofier Madeleine Flaucher
5	**Inhalt**	Das Modul vermittelt Konzepte, Prinzipien, Modelle, Methoden und Techniken für die effektive Entwicklung von benutzerfreundlichen Mensch-Computer-Schnittstellen. Das Thema moderner Benutzungsschnittstellen wird dabei für klassische Computer aber auch für mobile Geräte, eingebettete Systeme, Automobile und intelligente Umgebungen betrachtet. Die folgenden Themen werden im Modul behandelt: • Einführung in die Grundlagen der Mensch-Computer-Interaktion, historische Entwicklung • Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme • Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers • Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides • Ein- und Ausgabegeräte, Entwurfsraum für interaktive Systeme • Analyse-, Entwurfs- und Entwicklungsmethoden und -werkzeuge für Benutzungsschnittstellen • Prototypische Realisierung und Implementierung von interaktiven Systemen, Werkzeuge • Architekturen für interaktive Systeme, User Interface Toolkits und Komponenten • Akzeptanz, Evaluationsmethoden und Qualitätssicherung Contents: The module aims to teach basic knowledge of concepts, principles, models, methods and techniques for developing highly user-friendly Human-Computer Interfaces. Beyond traditional computer systems, modern user interfaces are also discussed in the context of automobile and intelligent environments, mobile devices and embedded systems. This module addresses the following topics: • Introduction to the basics of Human-Computer Interaction • Design principles and models for modern user interfaces and interactive systems • Information processing of humans, perception, motor skills, properties and skills of the users

		• Interaction concepts, metaphors, standards, norms and style guides • In- and output devices, design space for interactive systems • Analysis-, design- and development of methodologies and tools for easy-to-use user interfaces • Prototypic implementation of interactive systems • Architectures for interactive systems, User Interface Toolkits and components • Acceptance, evaluation methods and quality assurance
6	**Lernziele und Kompetenzen**	• Studierende entwickeln ein Verständnis für Modelle, Methoden und Konzepte der Mensch-Computer-Interaktion. • Sie lernen verschiedene Ansätze für den Entwurf, die Entwicklung und Bewertung von Benutzungsschnittstellen kennen und verstehen deren Vor- und Nachteile. • Die Teilnahme an der Veranstaltung versetzt Studierende in die Lage, einen Entwicklungsprozess in der Mensch-Computer-Interaktion zu verstehen und umzusetzen. • Sie werden weiterhin in die Lage versetzt, dies vor dem Hintergrund der Informationsverarbeitungsfähigkeit, Wahrnehmung und Motorik des Benutzers zu gestalten. • Passende Methoden der Evaluation sowie Akzeptanz- und Qualitätssicherung werden erlernt. Learning Objectives and Competences: • Students develop an understanding of models, methods and concepts in the field of Human-Computer Interaction. • They learn different approaches for designing, developing and evaluating User Interfaces and their advantages and disadvantages. • Joining the course enables students to understand and execute a development process in Human-Computer Interaction. • Students will be able to do a UI evaluation by learning the basics of information processing, perception and motoric skills of the user. • Appropriate evaluation methods, as well as acceptance and quality assurance aspects, will be learned.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 6
9	**Verwendbarkeit des Moduls**	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	elektronische Prüfung Electronic exam (in presence), 90min
11	**Berechnung der Modulnote**	elektronische Prüfung (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** 792501	**Forensische Informatik** Forensic computing (lecture with tutorial)	**5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr.-Ing. Felix Freiling
5	**Inhalt**	Forensische Informatik befasst sich mit der Sammlung, Aufbereitung und Analyse digitaler Beweismittel zur Verwendung vor Gericht. Ausgangspunkt ist jeweils der Verdacht auf einen Computereinbruch oder eine Straftat, die mit Hilfe von digitalen Geräten vorgenommen worden ist. Dieses Modul gibt einen Überblick über die Methoden der forensischen Informatik aus einer wissenschaftlichen Perspektive. Der Schwerpunkt liegt auf der Analyse von Dateisystemen. Ziel der Lehrveranstaltung ist nicht die Ausbildung von Forensik-Praktikern, sondern die Vermittlung von Kenntnissen, die es einem erlauben, Forschung im Bereich Computerforensik zu betreiben. Im Rahmen der Übung werden die Themen der Vorlesung im Rahmen von Fallstudien praktisch eingeübt. Voraussichtliche Themen: • Definition forensische Informatik • Der forensische Prozess und seine wissenschaftliche Fundierung • Rechtliche Rahmenbedingungen • Sichern von Festplatten • Analyse verschiedener Dateisysteme (FAT32, NTFS, Ext2/ Ext3) • Tools
6	**Lernziele und Kompetenzen**	Die Studierenden können Termini und Methoden der digitalen Forensik in die Entwicklung der forensischen Wissenschaften einordnen. Die Studierenden können die wesentlichen Datenstrukturen verschiedener Dateisysteme erklären. Sie können die für forensische Zwecke wesentlichen Datenstrukturen lokalisieren und geeignete Werkzeuge zu ihrer Analyse auswählen und anwenden. Die Studierenden können digitale Spuren konkreter Fallkonstellationen durch Anwendung von Werkzeugen rekonstruieren, analysieren, interpretieren und dokumentieren. Sie lernen ihre Untersuchungsergebnisse zu präsentieren und gegenüber kritischen Nachfragen zu verteidigen.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 6

9	**Verwendbarkeit des Moduls**	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	mündlich (30 Minuten) Die mündliche Prüfung dauert 30 Minuten.
11	**Berechnung der Modulnote**	mündlich (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 90 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Brian Carrier: File System Forensic Analysis. Addison-Wesley, 2005. • Eoghan Casey: Digital Evidence and Computer Crime - Forensic Science, Computers and the Internet, 3rd Edition. Academic Press 2011. • Andreas Dewald, Felix Freiling: Forensische Informatik. 3. Auflage, BoD, 2015.

1	**Modulbezeichnung** 93000	**Algorithmik kontinuierlicher Systeme** Algorithms for continuous systems	**7,5 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

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

		• berechnen iterativ Lösungen von nichtlinearen Gleichungen Analysieren Die Studierenden • klassifizieren Optimierungsprobleme • erforschen lineare Ausgleichsprobleme Lern- bzw. Methodenkompetenz Die Studierenden • ° lösen Aufgaben der Algorithmen kontinuierlicher Probleme in Gruppenarbeit
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4
9	**Verwendbarkeit des Moduls**	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten) Übungsleistung in den Übungen müssen 5 Theorieaufgaben und 5 Programmieraufgaben bearbeitet werden. Es müssen jeweils 50% der möglichen Punkte erreicht werden.
11	**Berechnung der Modulnote**	Klausur (100%) Übungsleistung (0%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

1	**Modulbezeichnung** 93121	**Theorie der Programmierung** Theory of 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**	Prof. Dr. Lutz Schröder
5	**Inhalt**	• Termersetzungssysteme, Normalisierung, Konfluenz • Getypter und ungetypter Lambda-Kalkül • Semantik von Programmiersprachen, Anfänge der Bereichstheorie • Datentypen, Kodatentypen, Induktion und Koinduktion, Rekursion und Korekursion • Programmverifikation, Floyd-Hoare-Kalkül • Reguläre Sprachen und endliche Automaten • Beschriftete Transitionssysteme, Bisimulation und Temporallogik
6	**Lernziele und Kompetenzen**	Fachkompetenz Wissen Die Studierenden geben elementare Definitionen und Fakten zu den behandelten Formalismen wieder. Verstehen Die Studierenden • erläutern Grundbegriffe der Syntax und Semantik von Formalismen und setzen diese zueinander in Bezug • beschreiben und erklären grundlegende Algorithmen zu logischem Schließen und Normalisierung • beschreiben wichtige Konstruktionen von Modellen, Automaten und Sprachen Anwenden Die Studierenden • verfassen formale Spezifikationen sequentieller und nebenläufiger Programme • verifizieren einfache Programme gegenüber ihrer Spezifikation durch Anwendung der relevanten Kalküle • ° führen einfache Beweise über Programme mittels Induktion und Koinduktion Analysieren Die Studierenden wählen für gegebene Verifikationsprobleme geeignete Formalismen aus erstellen einfache Meta-Analysen formaler Systeme, etwa Konfluenzprüfung von Termersetzungssystemen - führen einfache Meta-Beweise über Formalismen mittels Induktion und Koinduktion Lern- bzw. Methodenkompetenz Die Studierenden beherrschen das grundsätzliche

		Konzept des Beweises als hauptsächliche Methode des Erkenntnisgewinns in der theoretischen Informatik. Sie überblicken abstrakte Begriffsarchitekturen. Sozialkompetenz Die Studierenden lösen abstrakte Probleme in kollaborativer Gruppenarbeit.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 4;5;6
9	**Verwendbarkeit des Moduls**	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Klausur (90 Minuten)
11	**Berechnung der Modulnote**	Klausur (100%)
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 135 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	• Glynn Winskel, Formal Semantics of Programming Languages, MIT Press, 1993 • Michael Huth, Mark Ryan, Logic in Computer Science, Cambridge University Press, 2. Auflage 2004 • Henk Barendregt, The lambda-Calculus: Its Syntax and Semantics, North Holland, 1984 • John E. Hopcroft, Jeffrey D. Ullman and Rajeev Motwani, Introduction to Automata Theory, Languages, and Computation, 3rd ed., Prentice Hall, 2006 • Franz Baader, Tobias Nipkow, Term Rewriting and All That, Cambridge University Press, 1999

1	**Modulbezeichnung** 64640	**Mathematik für Naturwissenschaftler** Mathematics for natural scientists	**5 ECTS**
2	Lehrveranstaltungen	Vorlesung mit Übung: Mathematik für Naturwissenschaften (4 SWS) Übung: Übungen zur Mathematik für Naturwissenschaften (2 SWS)	- -
3	Lehrende	Dr. Alexander Prechtel	

4	**Modulverantwortliche/r**	Dr. Alexander Prechtel
5	**Inhalt**	• Grundbegriffe der linearen Algebra und Analysis • Komplexe Zahlen • Lineare Abbildungen, Matrizen, Gauss-Algorithmus, Determinanten, Eigenwerte und Eigenvektoren, Diagonalisierung • Stetige und differenzierbaren Funktionen, Taylor-Reihen, Integralrechung • Stabilitätsanalyse linearer Differentialgleichungssysteme Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.
6	**Lernziele und Kompetenzen**	Die Studierenden • definieren und erklären Grundbegriffe der Analysis und linearen Algebra; • verwenden grundlegende Verfahren und Algorithmen; • diskutieren Funktionen, Folgen und Reihen; • sammeln relevante Informationen, erkennen Zusammenhänge und bewerten diese.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 1
9	**Verwendbarkeit des Moduls**	Wahlpflichtbereich Informatik Gymnasium 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242 Zweitfachspezifische Module (LAGY INF) 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
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".

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	Linus Franke Prof. Dr. Marc Stamminger Laura Fink	

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 20242
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** 93197	**Praktikum: Entwicklung interaktiver eingebetteter Systeme** Laboratory course: Development of cyber-physical embedded systems	**10 ECTS**
2	Lehrveranstaltungen	Praktikum: Praktikum: Entwicklung interaktiver eingebetteter Systeme (8 SWS)	10 ECTS
3	Lehrende	Joachim Falk	

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

		Diese erzeugen im Anschluss Steuersignale für einen Motor, der einen verfahrbaren Schlitten antreibt.
6	**Lernziele und Kompetenzen**	Fachkompetenz - Wissen • Die Studierenden legen Konzepte des modellbasierten Entwurfs eingebetteter Systeme dar. Fachkompetenz - Verstehen • Die Studierenden veranschaulichen die Hauptaufgaben beim Systementwurf auf verschiedenen Abstraktionsebenen von der Anwendung selbst, dem Gesamtsystem, bis hinunter zu einem Hardwaremodul. • Die Studierenden schildern den Entwurf eines System von der Idee, über die Spezifikation bis zur Implementierung, der Analyse und letztendlich der Validierung an einem realen mechatronischen Versuchsaufbau. Fachkompetenz - Anwenden • Die Studierenden setzen die Integration von digitalen Hardware- und Software-Komponenten um. • Die Studierenden wenden die Programmiersprachen C/C++/ SystemC für die Entwicklung von Hardware- und Software-Komponenten an. Selbstkompetenz • Die Studierenden schätzen ihre individuellen Stärken ab, um einen geeignete Arbeitsaufteilung innerhalb der Gruppe festzulegen. Sozialkompetenz • Die Studierenden erarbeiten gemeinsam Schnittstellendefinitionen über Gruppengrenzen hinweg. • Die Studierenden erarbeiten kooperativ in Gruppen Lösungskonzepte und implementieren diese gemeinsam.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 7
9	**Verwendbarkeit des Moduls**	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%) Die Bewertung der Prüfungsleistung setzt sich zusammen aus Abschlussvortrag (Dauer: 25 min) und erfolgreicher Bearbeitung aller wöchentlichen Praktikumsaufgaben (verpflichtend, vor Ort an den Rechnerarbeitsplätzen des Lehrstuhls). Die Modulnote ergibt sich aus 50% Abschlussvortrag und 50% Mitarbeit im Praktikum basierend auf den abgegebenen Praktikumsaufgaben.
12	**Turnus des Angebots**	nur im Wintersemester

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

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

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

		Sozialkompetenz Lernende koennen in kleinen Teams effektiv und effizient zusammenarbeiten, die Arbeit gerecht verteilen und gemeinsam einen Bericht anfertigen.
7	**Voraussetzungen für die Teilnahme**	Grundkenntnisse in Python
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
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** 93129	**NWERC Praktikum** NWERC Training	**10 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

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

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

		• gehen mit Konflikten angemessen um und kommunizieren und handeln fair
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 7
9	**Verwendbarkeit des Moduls**	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung Das Praktikum setzt sich aus drei Komponenten zusammen: • Teamcontests (Juni bis Mitte November): Über die gesamte Praktikumsdauer werden voraussichtlich 25 Probe-Wettbewerbe (die genaue Anzahl kann erst zu Semesterbeginn festgelegt werden), zu je 5 Stunden abgehalten (insgesamt ca. 200 Aufgaben). Die Studierenden trainieren dabei in 3er-Teams mit wechselnder Besetzung. Nach jedem Probe-Wettbewerb werden die Lösungsansätze der gestellten Aufgaben besprochen. Danach haben die Teilnehmer zwei Wochen lang die Möglichkeit, eigenständig erarbeitete Lösungen für ungelöste Aufgaben auf der zur Verfügung gestellten Plattform (DOMJudge) einzureichen. Eine Aufgabe wird als gültig eingestuft, wenn sie alle Testfälle in einer vorgegebenen Zeit bestanden hat. Alle eingereichten Lösungsversuche werden sofort und automatisch evaluiert. Die Studierenden erhalten eine entsprechende Rückmeldung und es dürfen beliebig viele Lösungsversuche eingereicht werden. • Trainingslager (Anfang September): Anfang September findet ein 3-tägiges Trainingslager statt, bei dem anhand von weiteren Probe-Wettbewerben Teamstrategien trainiert werden. Die Übernachtungskosten werden von der Universität übernommen. Die Teilnehmer müssen nur für die Verpflegung aufkommen. Die Teilname am Trainingslager ist verpflichtend, um die Bewertung der Teamfähigkeit zu ermöglichen. • Einzelcontests (bis Mitte September): Nach dem Trainigslager finden zwei 5-stündige Einzelwettbewerbe statt, an denen die Studierenden individuell teilnehmen und bewertet werden. Wie bei den Teamcontests werden die Aufgaben nachbesprochen und die Studierenden haben die Möglichkeit, Lösungen innerhalb von zwei Wochen nachzureichen.
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	nur im Wintersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 90 h Eigenstudium: 210 h
14	**Dauer des Moduls**	1 Semester

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

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

4	**Modulverantwortliche/r**	Dr.-Ing. Vincent Christlein Felix Denzinger Fabian Wagner
5	**Inhalt**	At the Pattern Recognition Lab we offer practical topics that are connected to our current research in the fields of pattern recognition & machine learning, medical image processing, and big data applications. Other than a course with fixed topic, topics are defined individually. The 10 ECTS project is directed towards students of computer science. Also smaller 5 ECTS projects are available. The goal of this practical course is to familiarize the students with a Pattern Recognition System. Am Lehrstuhl für Mustererkennung bieten wir Praktika an, die mit unserer aktuellen Forschung in den Bereichen Mustererkennung & maschinelles Lernen, medizinische Bildverarbeitung und große Datenanwendungen verbunden sind. Im Gegensatz zu einem Kurs mit festem Thema werden die Themen individuell festgelegt. Das 10 ECTS-Projekt richtet sich an Studierende der Informatik. Auch kleinere 5 ECTS-Projekte sind möglich. Das Ziel dieses Praktikums ist es, die Studenten mit einem Mustererkennungssystem vertraut zu machen.
6	**Lernziele und Kompetenzen**	Die Studierenden • arbeiten sich in komplexe Softwaresysteme ein und erweitern diese • lernen, wie ein Mustererkennungssystem zu entwickeln ist • lernen, eigenständig Lösungsvorschläge auszuarbeiten und umzusetzen • dokumentieren die von ihnen geschriebene Software The students • familiarize themselves with complex software systems and extend them • learn to develop a pattern recognition system • learn to independently develop and implement proposed solutions • create documentation for the software written by them
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 7
9	**Verwendbarkeit des Moduls**	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung At the Pattern Recognition Lab we offer practical topics that are connected to our current research in the fields of pattern recognition &

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

1	**Modulbezeichnung** 93162	**Softwareentwicklungspraktikum Lehramt** Software development lab for teaching degree students	**10 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Prof. Dr. Marc-Pascal Berges
5	**Inhalt**	Agile Entwicklung eines Projektes/Programms für den Einsatz im Unterricht Organisation eines größeren Projektes Zeitmanagement Kommunikation in der Gruppe
6	**Lernziele und Kompetenzen**	Die Studierenden - entwickeln in Gruppen ein lauffähiges Programm für den Einsatz im Unterricht - verwalten ihren Projektfortschritt mit Hilfe eines Projektboards - lernen agile Entwicklung und deren Einsatzmöglichkeit im Unterricht kennen
7	**Voraussetzungen für die Teilnahme**	Es wird der erfolgreiche Abschluss der folgenden Module als Voraussetzung empfohlen: • Einführung in die Programmierung • Einführung in die Algorithmik • Einführung in Datenbanken • Einführung in das Software Engineering
8	**Einpassung in Studienverlaufsplan**	Semester: 7;5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung Prüfungsleistung: Entwicklung und Vorstellung eines Softwareprojektes in Gruppen. Die Vorstellung erfolgt in einem 20-30minütigem Vortrag. Die Entwicklung wird durch einen 30-60 seitigen Praktikumsbericht dokumentiert.
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	in jedem Semester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 120 h Eigenstudium: 180 h
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Best Practice Software-Engineering, A.Schatten, Spektrum Verlag

		Software-Qualität, Dirk W. Hoffmann, Springer Verlag

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

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

9	**Verwendbarkeit des Moduls**	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
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** 278855	**Praktikum Lego Mindstorms** Laboratory course: Lego mindstorms	**10 ECTS**
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	**Modulverantwortliche/r**	Joachim Falk Dr.-Ing. Stefan Wildermann
5	**Inhalt**	Informatik befasst sich nicht nur mit der Programmierung von Desktop-Rechnern. Vielmehr können Computer in immer mehr Gegenständen unseres Alltags oder in bestimmten technischen Kontexten gefunden werden. Man spricht hier von eingebetteten Systemen. Auch Roboter stellen solche eingebetteten Systeme dar. Ein Roboter erwacht durch sein Programm zum Leben. Die Programmierung von Robotern stellt einerseits eine Herausforderung dar. Andererseits ist sie aber auch mit viel Spaß verbunden. In diesem Modul werden LEGO Mindstorms Roboter verwendet, die mittels der Sprache Java programmiert werden. Dazu wird das Betriebssystem leJOS verwendet. Ziel des Moduls ist es, die Teamfähigkeit weiterzuentwickeln, indem ein praktisches Thema als Gruppe bearbeitet wird, Kenntnisse und Fertigkeiten in der Projektorganisation zu erwerben und die Fähigkeit der Problemlösung zu schulen.
6	**Lernziele und Kompetenzen**	**Fachkompetenz - Erschaffen** • Die Studierenden erstellen Lösungsideen für die Projekte und implementieren diese in Java für Lego Mindstorms Roboter. **Selbstkompetenz** • Die Studierenden schätzen ihre Stärken ab, um eine geeignete Arbeitsaufteilung innerhalb der Gruppe zu finden. **Sozialkompetenz** • Die Studierenden erarbeiten gemeinsam Projektpläne und -dokumentation im Themengebiet Robotik. • Die Studierenden organisieren selbstständig die gemeinsame Durchführung des Projekts und führen diese kooperativ in Gruppen durch.
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 7
9	**Verwendbarkeit des Moduls**	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)

		Die Bewertung der Prüfungsleistung setzt sich zusammen aus erfolgreicher Umsetzung einer Projektaufgabe, einem Abschlussvortrag (Dauer: ca. 25 min) und einem Praktikumsbericht (ca. 12 Seiten). Die Modulnote ergibt sich aus 20% Abschlussvortrag, 40% Praktikumsbericht und 40% Bewertung der umgesetzten Projektaufgabe.
12	**Turnus des Angebots**	nur im Sommersemester
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: 60 h Eigenstudium: 240 h
14	**Dauer des Moduls**	1 Semester
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	Weitere Informationen: https://www.cs12.tf.fau.de/lehre/lehrveranstaltungen/praktika/lego-mindstorms-praktikum/

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

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
10	**Studien- und Prüfungsleistungen**	Praktikumsleistung
11	**Berechnung der Modulnote**	Praktikumsleistung (100%)
12	**Turnus des Angebots**	keine Angaben zum Turnus des Angebots hinterlegt!
13	**Arbeitsaufwand in Zeitstunden**	Präsenzzeit: ?? h (keine Angaben zum Arbeitsaufwand in Präsenzzeit hinterlegt) Eigenstudium: ?? h (keine Angaben zum Arbeitsaufwand im Eigenstudium hinterlegt)
14	**Dauer des Moduls**	?? Semester (keine Angaben zur Dauer des Moduls hinterlegt)
15	**Unterrichts- und Prüfungssprache**	Deutsch
16	**Literaturhinweise**	

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

4	**Modulverantwortliche/r**	
5	**Inhalt**	keine Inhaltsbeschreibung hinterlegt!
6	**Lernziele und Kompetenzen**	keine Beschreibung der Lernziele und Kompetenzen hinterlegt!
7	**Voraussetzungen für die Teilnahme**	Keine
8	**Einpassung in Studienverlaufsplan**	Semester: 5
9	**Verwendbarkeit des Moduls**	Praktikum Informatik 1. Staatsprüfung für das Lehramt an Gymnasien Informatik 20242
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**	