UNIVERSITÄT BAYREUTH

Modulhandbuch

für Informatik LA Real
in der Fassung vom 20. Mai 2026

Fakultät für Mathematik, Physik und Informatik
Universität Bayreuth

Allgemeine Informationen und Lesehinweise

Ein zentraler Baustein des Bologna-Prozesses ist die Modularisierung der Studiengänge, das heißt die Umstellung des vormaligen Lehrveranstaltungssystems auf ein Modulsystem, in dem die Lehrveranstaltungen zu thematisch zusammenhängenden Veranstaltungsblöcken – also Modulen – gebündelt sind.

Dieses Modulhandbuch enthält die Beschreibungen aller Module, die im Studiengang angeboten werden. Das Modulhandbuch dient der Transparenz und versorgt Studierende, Studieninteressierte und andere interne und externe Adressatinnen und Adressaten mit Informationen über die Inhalte der einzelnen Module, ihre Qualifikationsziele sowie qualitative und quantitative Anforderungen.

Inhaltsverzeichnis und Index

Das Inhaltsverzeichnis gibt einen Überblick über die Modulbereiche und Module des Studiengangs. Die Klammerangabe nach dem Titel eines Moduls enthält den Zeitpunkt, zu dem dessen Beschreibung zuletzt aktualisiert wurde. Beispiel zur Notation: 24W steht für das Wintersemester 2024/25, 25S für das darauffolgende Sommersemester 2025.

Der Index am Ende des Modulhandbuchs listet alle Module des Studiengangs in alphabetischer Reihenfolge auf.

Modulbeschreibung

Die Beschreibung eines Moduls enthält dessen Lernziele, Lerninhalte und Leistungsnachweise. Sieht ein Modul mehrere Leistungsnachweise vor, ist deren Anteil an der Modulgesamtnote als Gewichtung angegeben. Eine detaillierte Beschreibung der Prüfungsformen, durch die die Leistungsnachweise erbracht werden, findet sich in § 9 der Allgemeinen Prüfungs- und Studienordnung für die Bachelor- und Masterstudiengänge an der Universität Bayreuth.

Durch Scannen oder Anklicken des QR-Code in der Beschreibung gelangt man zur Webseite des Moduls. Hierüber lassen sich die zum Modul gehörigen Lehrveranstaltungen einsehen und es wird angegeben, in welchen weiteren Studiengängen das Modul vorkommt.

Rechtsverbindlichkeit

Modulbeschreibungen dienen der Erhöhung der Transparenz und der besseren Orientierung über die Module eines Studiengangs. Rechtsverbindlich ist ausschließlich die einschlägige *(Fach-)Prüfungs- und Studienordnung.*

Inhaltsverzeichnis

Alphabetisches Verzeichnis befindet sich auf Seite 26

Fachwissenschaft (FW)
Fak127933: Konzepte der Programmierung (25W) ... 4
Fak127934: Rechnerarchitektur und Rechnernetze (25W) ... 5
Fak127935: Algorithmen und Datenstrukturen I (25W) ... 6
Fak116617: Theoretische Informatik für das Unterrichtsfach Informatik (17S) ... 7
Fak127951: Datenbanken und Informationssysteme I (25W) ... 8
Fak127952: Software Engineering (25W) ... 9
Fak112537: LAI 911: Programmierpraktikum (16S) ... 12
Fak112539: Softwarepraktikum für Lehramtsstudierende (20W) ... 13
Fak115494: LAI 941 - Seminar in Informatik (20W) ... 14
Wahlmodul aus INF 1xx/2xx/3xx
Fak127936: Betriebssysteme (25W) ... 15
Fak127953: Multimediale Systeme I (25W) ... 16
Fak127973: Sicherheit in verteilten Systemen (25W) ... 17
Fak127959: Computergraphik I (25W) ... 18
Fak127960: Datenbanken und Informationssysteme II (25W) ... 19
Fak610135: Pro/ENGINEER (14W) ... 20
Fachdidaktik (FD)
Fak114692: LAI 101: Informatik – Lehren und Lernen (15S) ... 21
Fak114698: LAI 401: Informatische Inhalte unter didaktischen Aspekten (15S) ... 22
Fak114699: LAI 402: Unterrichtspraxis Informatik (15S) ... 23
Fak114700: LAI 403: Schulpraktikum Informatik (15S) ... 24
Fak114693: LAI 102: Wahlmodul Didaktik der Informatik (15S) ... 25

Fak127933: Konzepte der Programmierung

Gültig ab: 01.10.2025

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Einsemestrig	90	150

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
8	Wintersemester	Mayer, Ruben; Prof. Dr.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Klausur oder mündliche Prüfung	1	8

Studien-/Prüfungsleistungen: Aktive Teilnahme an den Veranstaltungen und Selbststudium

Lernziele

Ziel der Veranstaltung ist, den Studierenden ein fundiertes Verständnis der Programmierung zu vermitteln, das im weiteren Studium als Fundament für die Informatik-Ausbildung dient. Dabei dient Java als Beispielsprache.
Der Vorkurs richtet sich insbesondere an Studierende ohne Programmiervorkenntnisse, wird aber allen Studierenden empfohlen. Er dient dazu, der fachlichen Diversität der Studierenden zu begegnen. Als Einstiegssprache wird Python verwendet.
Der Schwerpunkt liegt auf dem Erwerb von methodischen Kompetenzen: Durch das Verständnis fundamentaler Konzepte wie Kontroll- und Datenstrukturen, Methoden, Objektorientierung, Syntax, Typkonzept etc. sollen die Studierenden in die Lage versetzt werden, diese Konzepte bei der Umsetzung von Algorithmen in Programme einzusetzen und sich ferner in andere Programmiersprachen einzuarbeiten. Erste algorithmische Kompetenzen werden ebenfalls erworben. Sie legen die Grundlage für weiterführende Veranstaltungen (z.B. Algorithmen und Datenstrukturen).
Die Intensivübungen richten sich an Studierende, die die Modulprüfung im Sommersemester absolvieren und zur individuellen Förderung und Differenzierung.

Lerninhalte

Vorkurs: Grundbegriffe, Algorithmen, einfache Programme in Python
Einführung: Grundbegriffe
Algorithmen: wesentliche Eigenschaften, erste Beispiele
Programme: Umsetzung von Algorithmen in eine Programmiersprache
Syntax: EBNF, Ableitungsbäume, Syntaxdiagramme
Elementare Datentypen: ganze Zahlen, Gleitpunktzahlen, Zeichen, Wahrheitswerte
Ausdrücke: Syntax, Prioritäten, Auswertungsbäume
Anweisungen: Zuweisungen, Kontrollstrukturen, Flussdiagramme, strukturierte Programmierung
Methoden: Syntax, Aufruf, Ausführung, Aufrufstapel
Rekursion: Klassifikation von Rekursionsarten, Ausführung, Elimination von Rekursion
Strukturierte Datentypen: Arrays und Verbunde; Eigenschaften von Objekttypen, Referenzen
Objekte und Klassen: Grundbegriffe, Felder, Methoden, Konstruktoren, Klassen- vs. Objekteigenschaften, abstrakte Klassen
Vererbung: Einfachvererbung, Substituierbarkeit, Polymorphie, Vererbungsregeln
Schnittstellen: Konzept, Abgrenzung gegen abstrakte Klassen, Implementierung von Schnittstellen, Mehrfachvererbung auf Schnittstellen
Generizität: generische Datentypen, Abgrenzung von Generizität und Vererbung
Ausnahmebehandlung: Ausnahmeobjekte, Ausnahmebehandler, geschützte Blöcke
Funktionale Programmierung in Java
Inhalt der Intensivübung: Programmierung ausgewählter Kapitel der Vorlesung

Art und Umfang der Lehrveranstaltungen

Vorlesung (4 SWS)
Übungen (2 SWS)
Intensivübung (2 SWS, optional)
Vorkurs (1 SWS, optional)

Literatur:

K. Echtle, M. Goedicke: Lehrbuch der Programmierung, dpunkt.verlag, Heidelberg, 2000
H.P. Mössenböck: Sprechen Sie Java?, 5. Auflage, dpunkt.Verlag, Heidelberg, 2014
D.J. Barnes, M. Kölling: Objektorientierte Programmierung mit Java - Eine praxisnahe Einführung mit BlueJ, Pearson Studium, München (2003)
H. Balzert: Objektorientierte Programmierung mit Java 5, Spektrum Akademischer Verlag, Heidelberg (2005)
R. Sedgewick, K. Wayne: Einführung in die Programmierung mit Java, Pearson, München, 2011

zum Inhaltsverzeichnis

Fak127934: Rechnerarchitektur und Rechnernetze

Gültig ab: 01.10.2025

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Einsemestrig	90	150

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
8	Wintersemester	Rauber, Thomas; Prof. Dr.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Klausur oder mündliche Prüfung	9	7,2
semesterbegleitende Aufgaben	1	0,8

Lernziele

Das Ziel der Veranstaltung besteht in der Vermittlung grundlegender technologischer Kompetenz mit dem Schwerpunkt der Vermittlung von Kenntnissen des Aufbaus von Rechnersystemen mit Speicherhierarchie und Prozessoren. Vermittelt werden auch formale und algorithmische Kompetenzen, die zur Analyse und dem Entwurf digitaler Schaltkreise befähigen, sowie Design- und Realisierungskompetenzen zum Entwurf komplexer Schaltkreise. Durch Erlernen qualitativer Analyseverfahren zur Leistungsbewertung von Rechnersystemen und Rechnernetzen werden grundlegende methodische Kompetenzen im Bereich Rechnersysteme und Rechnernetze erworben, die Grundlagen für weiterführende Veranstaltungen legen.

Lerninhalte

Leistungsbewertung von Rechnern und grundsätzlicher Rechneraufbau
Maschinensprachen als Schnittstelle zwischen Hardware und Software
Zahlendarstellungen und Rechnerarithmetik
Entwurf digitaler Schaltkreise
Kombinatorische Schaltungen
Konstruktion von Speicherelementen
Speicherorganisation und Prozessorganisation,
Grundlagen und Leistungsbewertungen von Rechnernetzen
Schichtenprotokolle und Kommunikationsablauf
Wichtige Protokolle von Verbindungsschicht, Netzwerkschicht und Transportschicht

Art und Umfang der Lehrveranstaltungen

Vorlesung (4 SWS)
Übungen (2 SWS)

Literatur:

Patterson/Hennessy: Computer Organization & Design, Morgan Kaufmann, 6. Auflage, 2020
Hennessy/Patterson: Computer Architecture, Morgan Kaufmann, 7. Auflage, 2026
Oberschelp/Vossen: Rechneraufbau und Rechnerstrukturen, Oldenbourg Verlag, 10. Auflage, 2006
Kurose/Ross: Computer Networking, Addison Wesley, 8. Auflage, 2021
Peterson, Davie: Computer Networks, Morgan Kaufmann, 6. Auflage, 2021.

zum Inhaltsverzeichnis

Fak127935: Algorithmen und Datenstrukturen I

Gültig ab: 01.10.2025

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Einsemestrig	90	150

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
8	Sommersemester	Knauer, Christian; Prof. Dr.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Klausur	1	8

Studien-/Prüfungsleistungen: Aktive Teilnahme an den Veranstaltungen und Selbststudium

Voraussetzungen

Inhalte der Module
- Konzepte der Programmierung
- Formale Grundlagen der Informatik

Lernziele

Die Studenten sollen lernen, Daten zu strukturieren und dynamisch zu repräsentieren. Wichtig ist hierbei die enge Verknüpfung dieser Datenstrukturen und der hierauf angewandten Algorithmen. Ein weiteres Ziel ist die Vermittlung von Kenntnissen zur Komplexitätsanalyse von Algorithmen (methodische Kompetenz).
In der Fragestunde können Lehrinhalte beim Dozenten spezifisch nachgefragt und individuell nachgearbeitet werden. Sie dient der individuellen Förderung.

Lerninhalte

Listen, Keller, Such- und Sortierverfahren, binäre Bäume, Suchbäume (AVL), Graphen, Hash-Verfahren, Komplexität von Algorithmen, Algorithmentheorie.

Art und Umfang der Lehrveranstaltungen

Vorlesung (4 SWS)
Übung (2 SWS)
Fragestunde (1 SWS, optional)

Literatur:

Goodrich, Tamassia: “Data Structures & Algorithms in Java” (4th Ed.), 2006.
Saake, Sattler: “Algorithmen und Datenstrukturen – Eine Einführung mit Java” (3. Aufl.), 2006.
Cormen, Leiserson, Rivest, Stein: “Introduction to Algorithms” (2nd Ed.), 2001.

zum Inhaltsverzeichnis

Fak116617: Theoretische Informatik für das Unterrichtsfach Informatik

Gültig ab: 01.04.2017

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Leistungspunkte (LP) 8	**Turnus**	**Modulverantwortliche(r)** Martens, Wim; Prof. Dr.	

Leistungsnachweise

Titel:	Gewichtung:	LP:
Theoretische Informatik für das Unterrichtsfach Informatik	1	8

Lernziele

Die Studenten sollen die Grundlagen von regulären, kontextfreien, berechenbaren und effizient berechenbaren formalen Sprachen kennen und verstehen. Sie sollen in der Lage sein, Sprachen als regulär / nichtregulär und kontextfrei / nicht-kontextfrei einzuordnen und dies zu erklären.

Sie sollen in der Lage sein, die Verfahren aus der Vorlesung zu erklären und auf Beispiele anzuwenden.

zum Inhaltsverzeichnis

Fak127951: Datenbanken und Informationssysteme I

Gültig ab: 01.10.2025

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Einsemestrig	90	150

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
8	Sommersemester	Jablonski, Stefan; Prof. Dr.-Ing.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Klausur oder mündliche Prüfung	1	8

Studien-/Prüfungsleistungen: Aktive Teilnahme an den Veranstaltungen und Selbststudium

Lernziele

Ziel ist die Vermittlung grundlegender Kenntnisse zum Entwurf von (relationalen) Datenbanken. Die Studierenden sollen Analyse-, Entwurfs und Realisierungskompetenzen vermittelt bekommen, so dass sie selbständig eine Anwendungssituation analysieren und darauf aufbauend ein datenbankgestützte Anwendungen entwickeln können.
Daneben sollen Grundkenntnisse bezüglich des Aufbaus und des Betriebs von Datenbanksystemen vermittelt werden, so dass die Studierenden einen prinzipiellen Einblick in die Technologie von Datenbanksystemen bekommen.
Über den Übungsbetrieb sollen die Studierenden den praktischen Umgang mit Datenbanken und deren Anwendungen erlernen.
In den Intensivübungen werden darüber hinaus programmiertechnische Fähigkeiten vermittelt und Studierende individuell gefördert.

Lerninhalte

Entwurf von Datenbanksystemen: Aufbau konzeptioneller Schemata (Von Entity-Relationship-Diagrammen zu Relationen), Normalisierung, Relationenalgebra, Einführung in SQL, Verwendung von Datenbanksystemen (SQL als DB-Schnittstelle), Objektrelationale Datenbanksysteme; Aufbau von Datenbanksystemen (Architektur), Einführung ins Transaktionsmanagement; Aufbau von Informationssystemen (Arten von Informationssystemen), Anwendungen von Datenbanken in den Bereichen Bio-, Ingenieur- und Umweltinformatik; Vorstellung von Beispielen und Fallstudien.
Inhalt der Intensivübung: Programmierung ausgewählter Kapitel der Vorlesung.

Art und Umfang der Lehrveranstaltungen

Vorlesung (4 SWS)
Übung (2 SWS)
Intensivübung (2 SWS, optional)

Literatur:

Härder, T.; Rahm, E.: Architektur von Datenbanksystemen. Springer-Verlag
Gray, J.; Reuter, A.: Transaction Systems. Morgan Kaufman
Conolly, T.; Begg, C.: Database Systems: A Practical Approach to Design, Implementation and Management. Addison-Wesley Longman

zum Inhaltsverzeichnis

Fak127952: Software Engineering

Gültig ab: 01.10.2025

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Einsemestrig	90	150

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
8	Sommersemester	Baltes, Sebastian; Prof. Dr.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Klausur oder mündliche Prüfung zu Software Engineering	9	7,2
semesterbegleitende Aufgaben zu Software Engineering	1	0,8

Studien-/Prüfungsleistungen: Aktive Teilnahme an den Veranstaltungen und Selbststudium

Voraussetzungen

Inhalte der Module
- Konzepte der Programmierung
- Bachelor-Praktikum

Lernziele

Das Modul vermittelt ein systematisches und vertieftes Verständnis von fundamentalen Konzepten sowie ausgewählten Verfahren, Methoden und Werkzeugen des modernen Software-Engineerings. Die Studierenden sollen eine Auswahl an Verfahren, Methoden und Werkzeugen im Rahmen von Beispielen und Fallstudien anwenden. Insbesondere werden so Analyse- und Design-Kompetenzen vermittelt, die für die Entwicklung von Softwaresystemen von zentraler Bedeutung sind. Darüber hinaus werden methodische Kompetenzen unter anderem in Projektmanagement und Qualitätssicherung vermittelt.

Lerninhalte

Software Engineering: Geschichte, Begriffe, Programmieren im Kleinen vs. Programmieren im Großen, “No Silver Bullet”.

Software Development Processes: Software Development Life Cycle, historische Prozess-modelle (z.B., Wasserfallmodell, Spiralmodell), V-Modell, agile Prozessmodelle (Scrum, SA-Fe), DevOps.

Software Development Tools: Versionsverwaltungssysteme, kollaborative Softwareentwick-lung, integrierte Entwicklungsumgebungen, Build Automation, statische Analyse, Continuous Integration und Delivery, AI Coding Assistants.

Requirements Engineering: Requirements Elicitation, Analysis, Specification, Validation, Management; Stakeholder Groups, Anwendungsfälle, User Stories, Akzeptanzkriterien.

Software Architecture: Top-down vs. bottom-up Architecture, Module, Komponenten, Dekomposition, Architekturmuster, 4+1 Model.
Requirements to Code: Domain-driven design, Behavior-Driven Development, Test-Driven Development.

Software Documentation and Visualization: Dokumentationsökosysteme, Documentation as Code, Architectural Decision Records, Einführung in die Softwarevisualisierung.

Object-Oriented Modeling: Unified Modeling Language, Entwurfsmuster.

Software Quality: Antipatterns, Clean Code, funktionale vs. nicht-funktionale Qualität, Per-formance, Sicherheit, Usability.

Code Comprehension and Code Reviews: Mentale Modelle, kognitive Komplexität, lesba-ren Code schreiben, Coding Conventions, Kommentare im Code, Pull Request Reviews, Code Review Best Practices.

Software Testing: Begriffe, Regressionstests, Black-Box- vs. Glass-Box-Testing, risikoba-siertes Testen, Testbarkeit von Anforderungen, Testpyramide, manuelles vs. automatisiertes Testen, exploratives vs. strukturiertes Testen, Unit vs. Integration vs. System Testing, Code Coverage, Test Doubles, testbaren Code schreiben.

Managing Software Projects: Menschen managen, Risiken managen, Projekte managen, Aufwandsschätzung, Kanban Boards, Definition of Ready/Done, Gantt-Diagramme, Critical Path Method, Ethische Aspekte.

Open Source and Inner Source: Definition, Urheberrecht, Softwarelizenzen, Open-Source-Ökosysteme, Software Supply Chains, Open Source von und in Unternehmen.

Software Evolution and Maintenance: Evolution vs. Wartung, technische Schulden, De-bugging, Reverse-engineering, Reengineering, Refactoring, gute Bug Reports verfassen, Lehman’s Laws, Migrationen.

Software Measurement and Metrics: Softwaremetriken (insbesondere Komplexitätsmetri-ken), Geschäftsmetriken, Goodhart’s Law.

Cloud-native Software Development: Skalierbarkeit, Elastizität, Virtualisierung, Containe-rization.

Software Analytics: Einführung ins Empirical Software Engineering, Software Observability.

Art und Umfang der Lehrveranstaltungen

Vorlesung (4 SWS)
Übung (2 SWS)

Literatur:

Die Vorlesung basiert auf eigenen Materialien und einer Vielzahl von Quellen.
T. Winters, T. Manshreck, H. Wright: Software Engineering at Google - Lessons Learned from Programming Over Time, O'Reilly Media, 2020.
K. Wiegers, J. Beatty: Software Requirements, 3rd Edition, Microsoft Press, 2013.
P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, J. Stafford: Documenting Software Architectures - Views and Beyond, 2nd Edition, Addison-Wesley Professional, 2010.
M. Fowler: Patterns of Enterprise Application Architecture, Pearson International, 2002.
G. Hohpe. B. Woolf: Enterprise Integration Patterns - Designing, Building, and Deploying Messaging Solutions, Addison Wesley, 2003.
G. Booch, R.A. Maksimchuk. M.W. Engle, B.J. Young, J. Conallen, K.A. Houston: Object-Oriented Analysis and Design with Applications, Addison-Wesley Professional, 2007.
E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns - Elements of Reusable Object-Oriented Software, Addison-Wesley Professional, 1994.
M. Fowler: Refactoring - Improving the Design of Existing Code, 2nd Edition, Addison-Wesley Professional, 2018.
M. Aniche: Effective Software Testing - A Developer's Guide, Manning Publications, 2022.
E. Evans: A Domain-Driven Design - Tackling Complexity in the Heart of Software, Addison-Wesley Professional, 2003.
C. Sims, H.L. Johnson: Scrum - A Breathtakingly Brief and Agile Introduction, Dymaxicon, 2012.
R.C. Martin: Clean Code - A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008.
S. McConnell: Code Complete - A Practical Handbook of Software Construction, Microsoft Press, 2004.
Hunt, D. Thomas: The Pragmatic Programmer - From Journeyman to Master, Addison-Wesley Professional, 1999.
G. Hohpe: The Software Architect Elevator - Redefining the Architect's Role in the Digital Enterprise, O'Reilly Media, 2020.
J. Ousterhout: A Philosophy of Software Design, Yaknyam Press, 2021.

zum Inhaltsverzeichnis

Fak112537: LAI 911: Programmierpraktikum

Gültig ab: 01.04.2016

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Einsemestrig		

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
5	Sommersemester	Korch, Matthias; apl. Prof. Dr.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Programmierpraktikum	1	5

150 h gesamt (60 h Präsenz, 90 h Vorbereitung und Übungsbearbeitung)

Voraussetzungen

Keine

Lernziele

Vermittlung grundlegender Programmiertechniken; Einführung in die funktionale Programmierung; Einführung in die Programmierung mit Skriptsprachen

Lerninhalte

Vermittlung grundlegender Programmiertechniken; Einführung in die funktionale Programmierung; Einführung in die Programmierung mit Skriptsprachen

Art und Umfang der Lehrveranstaltungen

Praktikum

zum Inhaltsverzeichnis

Fak112539: Softwarepraktikum für Lehramtsstudierende

Gültig ab: 01.10.2020

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Einsemestrig	60	120

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
7	jedes Semester	Ehmann, Matthias; Dr.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Softwarepraktikum für Lehramtsstudierende	1	7

Modulprüfung

Voraussetzungen

INF 107 - Konzepte der Programmierung
INF 109 - Algorithmen und Datenstrukturen I

Lernziele

Die Studierenden entwickeln unter Anleitung kleinere Softwaresysteme in kleinen Gruppen. Im Vordergrund steht der Erwerb von algorithmischen, Design- und Realisierungskompetenzen. Die Studierenden sollen in die Lage versetzt werden, Software beschränkten Umfangs und beschränkten Schwierigkeitsgrads syste-matisch zu entwickeln (methodische Kompetenz) sowie die von ihnen erarbeitete Lösung zu präsentieren (kommunikative Kompetenz).

Lerninhalte

Probleme werden analysiert, Anforderungen definiert, ein Systementwurf erstellt, und die Komponenten des Systementwurfs werden implementiert und getestet. Hinzu kommt die Präsentation der Lösungskonzepte.

Anmerkungen

Präsentation der Aufgabenstellung und der Lösungskonzepte

Literatur:

Wolfram E. Rossig: Wissenschaftliches Arbeiten, Rossig Verlag, 6. Auflage, 2006

zum Inhaltsverzeichnis

Fak115494: LAI 941 - Seminar in Informatik

Gültig ab: 01.10.2020

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Zweisemestrig	30	120

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
3	Sommersemester	Ehmann, Matthias; Dr.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Seminar	1	3

Leistungsnachweis: Modul muss erfolgreich abgelegt werden; Bewertung geht nicht in die Gesamtnote des Studiums ein.

Voraussetzungen

INF 107 Konzepte der Programmierung
INF 109 - Algorithmen und Datenstrukturen I
INF 111: Theoretische Informatik I
INF 115: Software Engineering I

Lernziele

Erweiterung und Vertiefung der Inhalte aus den vorausgesetzten Modulen

Lerninhalte

Aufgaben aus den Gebieten Automatentheorie, formale Sprachen, Berechenbarkeitstheorie, Komplexitätsthe-orie, Datenstrukturen, Algorithmen und fortgeschrittener Programmierung, werden mit einer durch den Teil-nehmer erarbeiteten Lösung präsentiert und schriftlich ausgearbeitet.

Anmerkungen

"Multimedia-Präsentation" , schriftliche Ausarbeitung

Literatur:

Schöning: „Theoretische Informatik kurzgefasst“, 4. Aufl., 2001.
Saake, Sattler: “Algorithmen und Datenstrukturen – Eine Einführung mit Java” (3. Aufl.), 2006.

zum Inhaltsverzeichnis

Fak127936: Betriebssysteme

Gültig ab: 01.10.2025

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Einsemestrig	45	105

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
5	Wintersemester	Henrich, Dominik; Prof. Dr.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Klausur oder mündliche Prüfung zu Betriebssysteme	17	4,25
semesterbegleitende Aufgaben zu Betriebssyteme	3	0,75

Studienleistung: Aktive Teilnahme an den Veranstaltungen und Selbststudium
Prüfungsleistung: Siehe FPSO

Voraussetzungen

- Konzepte der Programmierung
- Rechnerarchitektur und Rechnernetze
- Algorithmen und Datenstrukturen I

Lernziele

Lernziele des Moduls sind das Verständnis des grundsätzlichen Aufbaus von Betriebssystemen, das Verständnis der eingesetzten Verfahren, sowie das Lernen der sinnvollen Auswahl und des Einsatz von Betriebssystemen. Es werden keine Einschränkung auf ein bestimmtes Betriebssystem vorgenommen und auch keine Implementierungsdetails vermittelt. Allgemein werden Methoden zur effizienten Verwaltung von zeitlichen bzw. räumlichen Ressourcen vermittelt.

Lerninhalte

Einleitung: Definition, Schnittstellen, Historie, Aufbau;
Prozessverwaltung: Prozesse/Threads, Prozesssynchronisation, -scheduling, -kommunikation;
Speicherverwaltung: Speicherbelegung, -adressierung, -seiten, Segmentierung, Caches, Schutz;
Dateiverwaltung: Dateisysteme, -namen, -attribute, -funktionen, -organisation;
Ein-/Ausgabeverwaltung: E/A-Aufgaben, Gerätemodelle, Treiber;
Systemsicherheit: Kryptographie, Authentifikation, Angriffe, Schutz;

Art und Umfang der Lehrveranstaltungen

Vorlesung (2 SWS)
Übung (1 SWS)

Anmerkungen

Medienformen: Multimedia-Präsentation als Vortrag und Ausdruck, Übungsblätter mit Korrektur, Tafelübungen

Literatur:

Tanenbaum A. S.: „Moderne Betriebssysteme“. 2. überarbeitete Auflage, München, Pearson Studium, 2002. –> 82/ST 260 T164 M6(3)+3. Dritte Auflage von 2009

zum Inhaltsverzeichnis

Fak127953: Multimediale Systeme I

Gültig ab: 01.10.2025

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Einsemestrig	45	105

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
5	Sommersemester	Guthe, Michael; Prof. Dr.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Klausur oder mündliche Prüfung	17	4,25
semesterbegleitende Aufgaben	3	0,75

Studien-/Prüfungsleistungen: Aktive Teilnahme an den Veranstaltungen und Selbststudium

Lernziele

Die Veranstaltung führt in die technologischen Grundlagen multimedialer Systeme ein. Im Mittelpunkt stehen Medientypen (Text, Bilder, Grafiken, 3D-Modelle, Audio und Video) und deren Repräsentation. Dabei werden jeweils grundlegende Standards zur Repräsentation von Medienobjekten besprochen. Darüber hinaus wird aber auch die werkzeugunterstützte Erstellung und Verwendung von Medienobjekten vermittelt.
In erster Linie dient die Veranstaltung dem Erwerb technologischer Kompetenzen.

Lerninhalte

Die Veranstaltung beginnt mit einer Einleitung über Medienobjekte, multimediale Objekte, multimediale Systeme und Medientypen. Darauf folgt ein Überblick über die verschiedenen Medientypen und Codierungen, angefangen mit der generellen Codierung von Text und Markup-Sprachen, über Grafik und Animation mit skalierbaren Vektorgrafiken (SVG) und VRML, Bildformate wie JPEG, JPEG 2000 und PNG, der digitalen Codierung von Audiosignalen mittels Psychoakustik (MP3 und AAC), bis hin zu Videoformaten, wie Analogvideo, HDTV, Digitalvideo, MPEG 1-4 und H.264.
Beispiele für Übungen werden so weit wie möglich aus den Anwendungsbereichen übernommen.

Art und Umfang der Lehrveranstaltungen

Vorlesung (2 SWS)
Übung (1 SWS)

Literatur:

Grauer M./Merten U.: Multimedia - Entwurf, Entwicklung und Einsatz in betrieblichen Informationssystemen, Springer-Verlag, Berlin/Heidelberg, 1997.
Henning Peter, A.: Taschenbuch Multimedia, 2. Auflage, Fachbuchverlag Leizig im Carl Hanser Verlag, 2001.
Henrich A.: Multimediatechnik, Skript zur Vorlesung, Universität Bamberg, 2004
Steinmetz R.: Multimedia-Technologie - Grundlagen, Komponenten und Systeme, (2., vollst. überarb. und erw. Aufl.), [u.a.]: Springer, 1999
F. Halsall: Multimedia Communications. Applications, Networks, Protocols and Standards, Addison Wesley, 2000.
A. Sloane: Internet Multimedia, Palgrave Macmillan, 2005.
T. Strutz: Bilddatenkompression. Grundlagen, Codierung, JPEG, MPEG, Wavelets, Vieweg Verlag, 2002, 2. Auflage.
J. Watkinson: The MPEG Handbook. MPEG-1, MPEG-2, MPEG-4 (MPEG-4 Part 10/H.264/AVC included), Focal Press, 2004; 2. Auflage.

zum Inhaltsverzeichnis

Fak127973: Sicherheit in verteilten Systemen

Gültig ab: 01.10.2025

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Einsemestrig	45	105

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
5	Unregelmäßig	Rauber, Thomas; Prof. Dr.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Klausur oder mündliche Prüfung	9	4,5
semesterbegleitende Aufgaben	1	0,5

Studien-/Prüfungsleistungen: Aktive Teilnahme an den Veranstaltungen und Selbststudium

Voraussetzungen

Inhalte der Module
- Konzepte der Programmierung
- Rechnerarchitektur und Rechnernetze

Lernziele

Ziel der Veranstaltung ist die Vermittlung grundlegender und vertiefender Kenntnisse wichtiger Techniken und Algorithmen, die die Sicherheit von Programmen in Netzwerkumgebungen gewährleisten. Dabei werden durch die Vermittlung grundlegender Sicherheitsaspekte in Softwaresystemen und Netzwerken analytische und methodische Kompetenzen erworben: die Studenten werden in die Lage versetzt, Softwaresysteme im Hinblick auf die Sicherheitsaspekte zu analysieren und geeignete Sicherheitstechniken zur Verbesserung der Sicherheit der Systeme einzusetzen.
Algorithmische und methodische Kompetenzen werden durch Vermittlung der methodischen Grundlagen von Verschlüsselungs- und Signaturtechniken und der darauf aufbauenden Algorithmen erworben.

Lerninhalte

- Sicherheitsprobleme in Programmen, Netzwerken und Netzwerkprotokollen
- Symmetrische und asymmetrische kryptographische Verfahren zur Verschlüsselung von Daten;
- Elektronische Signaturen und Schlüsselmanagement
Authentifizierungsverfahren: Grundlagen und Systeme
Firewall-Technologien und Sicherheitsprotokolle

Art und Umfang der Lehrveranstaltungen

Vorlesung (2 SWS)
Übung (1 SWS)

Literatur:

- Pfleeger: Security in Computing, Prentice Hall, 2003
- Bishop: Introduction to Computer Security, Addison Wesley, 2005
- Stallings: Cryptography and Network Security, 6. Auflage, Prentice Hall, 2013
- Eckert: IT-Sicherheit, Oldenbourg-Verlag, 9. Auflage, 2014

zum Inhaltsverzeichnis

Fak127959: Computergraphik I

Gültig ab: 01.10.2025

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch/Englisch	Einsemestrig	45	105

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
5	Sommersemester	Guthe, Michael; Prof. Dr.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Klausur oder mündliche Prüfung	17	4,25
semesterbegleitende Aufgaben	3	0,75

Studien-/Prüfungsleistungen: Aktive Teilnahme an den Veranstaltungen und Selbststudium

Voraussetzungen

Inhalte der Module
- Konzepte der Programmierung (oder vergleichbar)
- Algorithmen und Datenstrukturen I

Lernziele

Die Veranstaltung führt in die Grundlagen der interaktiven Grafikprogrammierung ein. Insbesondere sollen die Struktur von und der Umgang mit Graphikpaketen erlernt werden, sowie das Design und die Analyse von Algorithmen der Computergraphik. Insbesondere soll dabei auch die Shaderprogrammierung erlernt werden.

Lerninhalte

Der erste Teil der Veranstaltung behandelt die Graphik-Hardware und die Rendering Pipeline. Dazu gehören unter anderem die Repräsentation von Objekten, Raster- und Sichtbarkeitsalgorithmen sowie einfache Beleuchtungsmodelle und Texturen. Parallel dazu werden in den Übungen die Benutzung der OpenGL Rendering API und die Shaderprogrammierung mit Cg abgehandelt. Neben hardwarenahen API beschäftigt sich die Vorlesung im zweiten Teil mit Szenengraphen, wie sie in aktuellen Game-Engines verwendet werden.

Art und Umfang der Lehrveranstaltungen

Vorlesung (2 SWS)
Übung (1 SWS)

Literatur:

- P. Shirley: Fundamentals of Computer Graphics (2nd Edition), 2006, 2. Auflage
- D. Hearn, M. P. Baker: Computer Graphics with OpenGL (3rd Edition), 2003, 3. Auflage.
- R. Fernando, M.J. Kilgard: The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics, Addison Wesley Longman, 2003.
- J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes: Computer Graphics - Principles and Practice, Addison Wesley, 1990.
- J. Encarnacao; W. Straßer, R. Klein: Graphische Datenverarbeitung I und II, Oldenbourg, 1996.
- A. Watt: 3D Computer Graphics, Addison-Wesley 1999.
- A. Watt; F. Policarpo: 3D Games: Real-time Rendering and Software Technology, Addison Wesley 2001.

zum Inhaltsverzeichnis

Fak127960: Datenbanken und Informationssysteme II

Gültig ab: 01.10.2025

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Einsemestrig	45	105

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
5	Wintersemester	Jablonski, Stefan; Prof. Dr.-Ing.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Klausur oder mündliche Prüfung	1	5

Studien-/Prüfungsleistungen: Aktive Teilnahme an den Veranstaltungen und Selbststudium

Voraussetzungen

Inhalte des Moduls
- Datenbanken und Informationssysteme I

Lernziele

Vermittlung vertiefter technologischer Kenntnisse zur Umsetzung von Datenbanksystemen hinsichtlich Aufbau (Architektur) und Transaktionsmanagement; Vermittlung von analytischen Fähigkeiten zum Aufbau von Schichtenarchitekturen; Über den Übungsbetrieb werden Realisierungskompetenzen hinsichtlich der Umsetzung von komplexen Architekturen vermittelt. Auf die Umsetzung komplexer Architekturen in den Anwendungsgebieten wird eingegangen. In der Intensivübung wird im Rahmen der Vorbereitung auf die Modulprüfung durch individuelle Behandlung der Fragen von Studierenden der fachlichen Diversität begegnet.

Lerninhalte

Architektur von Datenbanksysteme: Externspeicherverwaltung, Systempufferverwaltung, Zugriffspfade, Seitenverwaltung, interne, satzorientierte und mengenorientierte Schnittstelle; Transaktionsverarbeitung: ACID-Konzept, Implementierung von transaktionalen Eigenschaften, Synchronisation, 2PC-Protokoll, Logging, Recovery, Transaktionsmodelle;
Anwendung von Architekturmodellen auf komplexe Anwendungen der Anwendungsbereiche.
Die Intensivübung greift wichtige Inhalte der Vorlesung auf und vertieft diese.

Art und Umfang der Lehrveranstaltungen

Vorlesung (2 SWS)
Übung (1 SWS)

Literatur:

Härder, T.; Rahm, E.: Architektur von Datenbanksystemen. Springer-Verlag
Gray, J.; Reuter, A.: Transaction Systems. Morgan Kaufman
Conolly, T.; Begg, C.: Database Systems: A Practical Approach to Design, Implementation and Management. Addison-Wesley Longman

zum Inhaltsverzeichnis

Fak610135: Pro/ENGINEER

Gültig ab: 01.10.2014

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Deutsch	Einsemestrig	60	0

Leistungspunkte (LP)	Turnus	Modulverantwortliche(r)
2	Wintersemester	Tremmel, Stephan; Prof. Dr.-Ing.

Leistungsnachweise

Titel:	Gewichtung:	LP:
Pro/ENGINEER	1	2

- Zweiwöchiger Blockkurs 60 Std.
Summe (2 ECTS) 60 Std.

- Mitarbeit im Blockkurs

Voraussetzungen

Keine

Lernziele

Grundverständnis für alle wichtigen Aufgaben und Arbeitsgebiete eines Ingenieurs; das Wissen und die Fähigkeiten, die ein Konstrukteur auf Sachbearbeiterebene braucht; Kenntnis bereichspezifischer Softwarewerkzeuge und Fähigkeit zu deren Anwendung.

Lerninhalte

Konstruktion und Berechnung von Maschinenelementen und daraus zusammengesetzter Maschinen; Aufgaben und Arbeitsgebiete eines Ingenieurs in der Konstruktion; Einführung in einfache Finite-Elemente-Berechnungen

Art und Umfang der Lehrveranstaltungen

2-wöchiger Blockkurs (entspricht 4 SWS)

zum Inhaltsverzeichnis

Fak114692: LAI 101: Informatik – Lehren und Lernen

Gültig ab: 01.04.2015

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Leistungspunkte (LP) 5	**Turnus**	**Modulverantwortliche(r)** Ehmann, Matthias; Dr.	

Leistungsnachweise

Titel:	Gewichtung:	LP:
Modulprüfung Informatik - Lehren und Lernen	1	5

Lernziele

Erläutern und Anwenden der Konzepte des Lehrens und Lernens im Fach Informatik Beherrschen der Denkweisen und Methoden der Informatik und ihre Übertragung auf den Schulunterricht Konzeption und Gestaltung von Informatikunterricht

zum Inhaltsverzeichnis

Fak114698: LAI 401: Informatische Inhalte unter didaktischen Aspekten

Gültig ab: 01.04.2015

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Leistungspunkte (LP) 5	**Turnus**	**Modulverantwortliche(r)** Ehmann, Matthias; Dr.	

Leistungsnachweise

Titel:	Gewichtung:	LP:
Modulprüfung Informatische Inhalte unter didaktischen Aspekten	1	5

Lernziele

Vorlesung: siehe Ausführungen bei den Wahlpflichtveranstaltungen im Modul LAI 301 Seminar: Erschließen geeigneter Lernfelder des Unterrichtsfaches Informationstechnologie an Realschulen

zum Inhaltsverzeichnis

Fak114699: LAI 402: Unterrichtspraxis Informatik

Gültig ab: 01.04.2015

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Leistungspunkte (LP) 5	**Turnus**	**Modulverantwortliche(r)** Ehmann, Matthias; Dr.	

Leistungsnachweise

Titel:	Gewichtung:	LP:
Modulprüfung Unterrichtspraxis Informatik	1	5

Lernziele

Entwicklung didaktischer Kompetenz bei der Planung von Informatikunterricht Einblick in Möglichkeiten der Evaluation von Lehr- und Lernprozessen im Informatikunterricht Einsatz und Anwendung von Informatiksystemen im Unterricht an Realschulen

zum Inhaltsverzeichnis

Fak114700: LAI 403: Schulpraktikum Informatik

Gültig ab: 01.04.2015

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Leistungspunkte (LP) 6	**Turnus**	**Modulverantwortliche(r)** Ehmann, Matthias; Dr.	

Leistungsnachweise

Titel:	Gewichtung:	LP:
Modulprüfung Schulpraktikum Informatik	1	6

Lerninhalte

Entwicklung didaktischer Kompetenz bei der Planung und Durchführung von Informatikunterricht Einblick in Möglichkeiten der Evaluation von Lehr- und Lernprozessen im Informatikunterricht

zum Inhaltsverzeichnis

Fak114693: LAI 102: Wahlmodul Didaktik der Informatik

Gültig ab: 01.04.2015

Lehrsprache	Dauer	Präsenzstudium (in Stunden)	Selbststudium (in Stunden)
Leistungspunkte (LP) 5	**Turnus**	**Modulverantwortliche(r)** Ehmann, Matthias; Dr.	

Leistungsnachweise

Titel:	Gewichtung:	LP:
Modulprüfung Wahlmodul Didaktik der Informatik	1	5

Lernziele

Wahlpflichtseminar: siehe Modul LAI 301 Kompaktkurs: Sicherer Umgang mit Medien im Informatikunterricht

zum Inhaltsverzeichnis

Index

Fak127935: Algorithmen und Datenstrukturen I (25W) ... 6
Fak127936: Betriebssysteme (25W) ... 15
Fak127959: Computergraphik I (25W) ... 18
Fak127951: Datenbanken und Informationssysteme I (25W) ... 8
Fak127960: Datenbanken und Informationssysteme II (25W) ... 19
Fak127933: Konzepte der Programmierung (25W) ... 4
Fak114692: LAI 101: Informatik – Lehren und Lernen (15S) ... 21
Fak114693: LAI 102: Wahlmodul Didaktik der Informatik (15S) ... 25
Fak114698: LAI 401: Informatische Inhalte unter didaktischen Aspekten (15S) ... 22
Fak114699: LAI 402: Unterrichtspraxis Informatik (15S) ... 23
Fak114700: LAI 403: Schulpraktikum Informatik (15S) ... 24
Fak112537: LAI 911: Programmierpraktikum (16S) ... 12
Fak115494: LAI 941 - Seminar in Informatik (20W) ... 14
Fak127953: Multimediale Systeme I (25W) ... 16
Fak610135: Pro/ENGINEER (14W) ... 20
Fak127934: Rechnerarchitektur und Rechnernetze (25W) ... 5
Fak127973: Sicherheit in verteilten Systemen (25W) ... 17
Fak127952: Software Engineering (25W) ... 9
Fak112539: Softwarepraktikum für Lehramtsstudierende (20W) ... 13
Fak116617: Theoretische Informatik für das Unterrichtsfach Informatik (17S) ... 7