



UNIVERSITÄT
DES
SAARLANDES

FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

MODULHANDBUCH

Lehramtsstudienfach Informatik

20. Juni 2025

Liste der Modulbereiche und Module

1	Grundmodule	3
1.1	Ausgewählte Themen für das Informatik-Lehramt (groß)	4
1.2	Ausgewählte Themen für das Informatik-Lehramt (klein)	6
1.3	Big Data Engineering	8
1.4	Datenbankentwurf und Managementsysteme	11
1.5	Datenschutzrecht	12
1.6	Einführung in die Didaktik der Informatik	13
1.7	Elementare fachdidaktische schulpraktische Studien	15
1.8	Fachdidaktikseminar I	17
1.9	Fachdidaktikseminar II	18
1.10	Grundzüge der Theoretischen Informatik	19
1.11	Informationsmanagement	21
1.12	Mathematik für Informatik-Lehramt 1	22
1.13	Mathematik für Informatik-Lehramt 2	24
1.14	Programmierung 1	26
1.15	Programmierung 2	28
1.16	Rechnergestütztes Controlling und Controlling mit SAP ERP	30
1.17	Software Engineering Lab (Softwarepraktikum) LA LAB und Sek I+II	31
1.18	Systemarchitektur für Informatik-Lehramt	33
1.19	Vertiefende fachdidaktische schulpraktische Studien	35
1.20	Webtechnologien	36
1.21	Zivilrechtliche Grundlagen des IT-Rechts	37
2	Wahlpflichtbereich 1	38
2.1	English for Physical Sciences	39
2.2	Perspektiven der Informatik	40
3	Wahlpflichtbereich 2	41
3.1	Elements of Data Science and Artificial Intelligence	42
3.2	Elements of Machine Learning	43

3.3	Foundations of Cybersecurity 1	45
3.4	Grundzüge von Algorithmen und Datenstrukturen	46
3.5	Nebenläufige Programmierung	47
4	Wahlpflichtbereich 3	49
4.1	Algorithms and Data Structures	50
4.2	Artificial Intelligence	51
4.3	Automated Reasoning	53
4.4	Compiler Construction	54
4.5	Complexity Theory	55
4.6	Computer Algebra	56
4.7	Computer Graphics	57
4.8	Cryptography	59
4.9	Cyber-Physical Systems	60
4.10	Data Networks	62
4.11	Database Systems	64
4.12	Digital Transmission & Signal Processing	66
4.13	Distributed Systems	68
4.14	Geometric Modelling	69
4.15	Human Computer Interaction	71
4.16	Image Processing and Computer Vision	72
4.17	Information Retrieval and Data Mining	74
4.18	Introduction to Computational Logic	75
4.19	Machine Learning	76
4.20	Operating Systems	77
4.21	Optimization	79
4.22	Security	80
4.23	Semantics	81
4.24	Software Engineering	82
4.25	Verification	84

Modulbereich 1

Grundmodule

Ausgewählte Themen für das Informatik-Lehramt (groß)

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

5-7

jährlich

1 Semester

3

7

Modulverantwortliche/r Prof. Dr. Verena Wolf

Dozent/inn/en Prof. Dr. Verena Wolf

Zulassungsvoraussetzungen Keine

Leistungskontrollen / Prüfungen

- Beiträge zur Diskussion
- Thematischer Vortrag
- Schriftliche Ausarbeitung

Lehrveranstaltungen / SWS Seminar 3 SWS (bis zu 20 Studierende)

Arbeitsaufwand 210h = 45 h Präsenz und 165 h Eigenstudium

Modulnote Wird aus Leistungen im Vortrag, Beiträgen zur Diskussion und der schriftlichen Ausarbeitung ermittelt.

Sprache Deutsch

Lernziele / Kompetenzen

Die Studierenden haben am Ende der Veranstaltung ein gutes Verständnis der behandelten Themen der Informatik erlangt und sind in der Lage Unterrichtskonzepte zu diesen Themen zu konzipieren.

Sie haben Kompetenz im Diskutieren, Kritisieren und Präsentieren von wissenschaftlichen Erkenntnissen gewonnen.

Inhalt

Es werden ausgewählte fachliche Themen mit Bezug zur Didaktik der Informatik behandelt. Lernziele

- Erwerb von zusätzlichem fachlichen Wissen zu den behandelten Themen
- Ausarbeitung didaktischer Konzepte für Unterrichtssequenzen
- Verfassen eigener Zusammenfassungen zu den behandelten fachlichen Themen
- Diskussion der Arbeiten in der Gruppe
- Präsentationstechnik

Themen:

- Grundlagen des maschinellen Lernens (Übersicht und Einordnung von Teilbereichen; generelle Vorgehensweise anhand von Entscheidungsbäumen; prinzipielle Funktionsweise von künstlichen neuronalen Netzen, einfache Clustering-Verfahren)
- Grundlagen von Datennetzen (Internetprotokolle, Schichtenmodell, DNS- und Webserver)
- Grundlagen der Computerethik (ethische Standards, Datenschutz und Anonymität im Internet, Computerkriminalität, Ethik für die künstliche Intelligenz)
Der typische Ablauf eines Seminars ist wie folgt:
- Vorbereitende Gespräche zur Themenauswahl
- Regelmäßige Treffen mit Diskussion ausgewählter Beiträge
- Vortrag und Ausarbeitung zu einem der Beiträge

Literaturhinweise

Wird kurzfristig vom Dozenten bekanntgegeben.

Ausgewählte Themen für das Informatik-Lehramt (klein)

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

5-7

jährlich

1 Semester

2

4

Modulverantwortliche/r Prof. Dr. Verena Wolf

Dozent/inn/en Prof. Dr. Verena Wolf

Zulassungsvoraussetzungen Grundlegende Veranstaltungen der Informatikdidaktik und fachliche Veranstaltungen (empfohlen)

Leistungskontrollen / Prüfungen

- Beiträge zur Diskussion
- Thematischer Vortrag

Lehrveranstaltungen / SWS Seminar 2 SWS (bis zu 20 Studierende)

Arbeitsaufwand

- 45 h Präsenzstudium
- + 75 h Eigenstudium
- = 120 h (= 4 ECTS)

Modulnote Wird aus Leistungen im Vortrag und Beiträgen zur Diskussion ermittelt.

Sprache Deutsch

Lernziele / Kompetenzen

Die Studierenden haben am Ende der Veranstaltung ein gutes Verständnis der behandelten Themen der Informatik erlangt und sind in der Lage Unterrichtskonzepte zu diesen Themen zu konzipieren. Der Schwerpunkt des Seminars liegt im selbstständigen Erarbeiten neuer Themenbereiche.

Sie haben Kompetenz im Diskutieren, Kritisieren und Präsentieren von wissenschaftlichen Erkenntnissen gewonnen.

Inhalt

Es werden ausgewählte fachliche Themen mit Bezug zur Didaktik der Informatik behandelt. Lernziele

- Erwerb von zusätzlichem fachlichen Wissen zu den behandelten Themen
- Ausarbeitung didaktischer Konzepte für Unterrichtssequenzen
- Diskussion der Arbeiten in der Gruppe
- Präsentationstechnik

Themen:

- Grundlagen des maschinellen Lernens (Übersicht und Einordnung von Teilbereichen, generelle Vorgehensweise anhand von Entscheidungsbäumen, prinzipielle Funktionsweise von künstlichen neuronalen Netzen, einfache Clustering-Verfahren)
- Grundlagen von Datennetzen (Internetprotokolle, Schichtenmodell, DNS- und Webserver)
- Grundlagen der Computerethik (ethische Standards, Datenschutz und Anonymität im Internet, Computerkriminalität, Ethik für die künstliche Intelligenz)
Der typische Ablauf eines Seminars ist wie folgt:
- Vorbereitende Gespräche zur Themenauswahl
- Regelmäßige Treffen mit Diskussion ausgewählter Beiträge
- Vortrag zu einem der Beiträge

Literaturhinweise

Wird kurzfristig vom Dozenten bekannt gegeben.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	4-8	jedes Sommersemester	1 Semester	4	6

Modulverantwortliche/r Prof. Dr. Jens Dittrich

Dozent/inn/en Prof. Dr. Jens Dittrich

Zulassungsvoraussetzungen *Programmierung 1, Programmierung 2, Softwarepraktikum oder Projektpraktikum, Mathematik für Informatiker 1, sowie Grundzüge von Algorithmen und Datenstrukturen (jeweils empfohlen).*

Leistungskontrollen / Prüfungen Erfolgreiche Teilnahme an den Übungen/Projekt berechtigt zur Teilnahme an der Abschlussklausur.

Lehrveranstaltungen / SWS 2 SWS Vorlesung
+ 2 SWS Übung
= 4 SWS

Arbeitsaufwand 60 h Präsenzstudium
+ 120 h Eigenstudium
= 180 h (= 6 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen, und ggf. Projekt ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekanntgegeben.

Sprache Englisch

Lernziele / Kompetenzen

Die Vorlesung vermittelt grundlegende Kenntnisse über fundamental Konzepte von Datenmanagement und Datenanalyse im Kontext von Big Data und Data Science.

Im Rahmen der Übungen kann während des Semesters ein durchgehendes Projekt durchgeführt. Dies kann zum Beispiel ein soziales Netzwerk (im Stil von Facebook) sein bzw. jedes andere Projekt, in dem Techniken des Datenmanagements eingeübt werden können (z.B. naturwissenschaftliche Daten, Bilddaten, andere Webapplikationen, etc.). Zunächst wird dieses Projekt in E/R modelliert, dann umgesetzt und implementiert in einem Datenbankschema. Danach wird das Projekt erweitert, um auch unstrukturierte Daten verwalten und analysieren zu können. Insgesamt werden so an einem einzigen Projekt alle fundamentalen Techniken gezeigt, die für das Verwalten und Analysieren von Daten wichtig sind.

Inhalt

1 Einführung und Einordnung

Einordnung und Abgrenzung: Data Science
Wert von Daten: Das Gold des 21. Jahrhunderts
Bedeutung von Datenbanksystemen
Architekturen: 2-Tier, 3-Tier, etc
Was sind eigentlich Daten?
Modellierung vs Realität
Kosten mangelhafter Modellierung
Datenbanksystem nutzen vs selbst entwickeln
Positive Beispiele für Apps
Anforderungen
Literaturhinweise
Vorlesungsmodus

- 2 Datenmodellierung
 - Motivation
 - E/R
 - Relationales Modell
 - Hierarchische Daten
 - Graphen und RDF
 - Redundanz, Normalisierung, Denormalisierung
 - Objektrelationale DBMS
- 3 Anfragesprachen
 - Relationale Algebra
 - Hierarchische Anfragesprachen
 - Graphorientierte Anfragesprachen
- 4 SQL
 - Grundlagen
 - Zusammenhang mit relationaler Algebra
 - PostgreSQL
 - Integritätsbedingungen
 - Transaktionskonzept
 - ACID
 - Sichten (und access control lists)
- 5 Implementierungstechniken
 - Übersicht
 - vom WAS zum WIE
 - Kosten verschiedener Operationen
 - EXPLAIN
 - Physisches Design
 - Indexe, Tuning
 - Datenbank-Tuning
 - Regelbasierte Anfrageoptimierung
 - Kostenbasierte Anfrageoptimierung
 - Machine Learning als Anfrageoptimierungstechnik
- 6 Zeitliche und räumliche Daten
 - als Teil des Schemas
 - as of/time travel
 - append-only und Streaming
 - Versioning
 - Snapshotting (Software und OS-basiert)
 - Differential Files/LSM et al
 - Publish/Subscribe
 - Indexstrukturen
- 7 Recovery, Durability, Archivierung
 - Grundproblematik
 - Vergessen vs Komprimieren vs Kondensieren
 - Heiße vs kalte Daten
 - Archivierung
 - Redundanz
 - Implementierungsaspekte
 - UNDO/REDO
 - Logging
- 8 Nebenläufigkeitskontrolle
 - Serialisierbarkeitstheorie
 - Isolationslevels
 - Verteilte Datenbanksysteme: Sharding, HP, VP, permissioned Blockchains
 - Implementierungsaspekte

9 ETL und Data Cleaning

- Datenbankschnittstellen: JDBC et al
- Textdatenbanken: CSV, SQLite
- Data Warehousing
- Schema Matching
- Reporting
- Data Cleaning
- Denormalisierung, Caching, Materialisierung
- Workflows
- ETL und Data Science in Data Science und Machine Learning

10 Big Data

- Was ist eigentlich Big Data?
- Big Data vs Privatheit
- Beispiele: Zusammenführen von Daten
- Physische Barrieren

11 NoSQL

- Key/Value Stores
- KeyDocument Stores: MongoDB
- MapReduce
- Flink
- Spark

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Dieses Modul wurde früher auch unter dem Namen *Informationssysteme* geführt. Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Big Data Engineering*.

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

4-8

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

1-5

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	3-7	i.d.R. jährlich	2-3 Semester	6	9

Modulverantwortliche/r Prof. Dr. Verena Wolf

Dozent/inn/en StD Tim Lethen

Zulassungsvoraussetzungen Keine

Leistungskontrollen / Prüfungen

- Beiträge zur Diskussion
- Thematische Vorträge
- Schriftliche Ausarbeitung
- Schriftliche Abschlussprüfung oder mündliches Abschlussgespräch
- Praktikumsbericht (Laborpraktikum)

Lehrveranstaltungen / SWS Vorlesung: 30 h Präsenz
60 h Selbststudium

Seminare/Praktikum: 60 h Präsenz
20 h Vor- und Nacharbeit

Arbeitsaufwand

- Einführung in die Didaktik der Informatik Vorlesung mit integrierten Übungen (2 SWS/ 3 KPe)
- Fachdidaktikseminar I Seminar zur fachdidaktischen Behandlung spezieller inhaltlicher Themen der Schulinformatik Seminar (2 SWS/ 3 KPe)
- Fachdidaktikseminar II (wie Fachdidaktikseminar I, jedoch zu anderen Themen) oder Fachdidaktisches Laborpraktikum Laborpraktikum zur Programmierung, Softwaretechnik oder zu Themen der Technischen Informatik (Robotik, Rechnerbau) (2 SWS, 3 KPe)

Modulnote Die Einzelnoten der Moduleile werden aus den Beurteilungen thematischer Vorträge und Projektarbeiten sowie den Noten in mündlichen oder schriftlichen Abschlussprüfungen ermittelt.

Die Modulnote ergibt sich aus dem arithmetischen Mittel der Noten der drei Moduleile.

Sprache Deutsch

Lernziele / Kompetenzen

Die Studierenden

- lernen die wichtigsten Konzepte informatischer Bildung kennen
- werden mit speziellen Fragen und Anforderungen des Informatikunterrichtes vertraut
- können Themen der Fachwissenschaft Informatik didaktisch aufbereiten
- lernen die Prinzipien didaktischen Handelns im Umfeld der Fachwissenschaft Informatik kennen
- werden mit den speziellen Methoden und Werkzeugen des Informatikunterrichtes vertraut
- erwerben Fähigkeiten zur Planung und Umsetzung von Informatikunterricht
- lernen die Lehrpläne des Faches Informatik an der Schulform ihrer Wahl kennen und erhalten einen Einblick in die Genese von Lehrplänen
- werden auf die speziellen Anforderungen des Informatikunterrichtes vorbereitet

Inhalt

- Themen, Inhalte und Ziele informatischer Bildung
- Grundlegende Konzepte und Inhalte von Informatikunterricht
- Methodik und Strategien der Unterrichtsorganisation
- Für den Informatikunterricht spezifische Lehr- und Lernformen
- Vorgehensweise bei der Aufbereitung und der unterrichtlichen Einbindung fachwissenschaftlicher Inhalte
- Ausarbeitung didaktischer Konzepte für Unterrichtssequenzen
- Fachspezifische Lehr- und Lernschwierigkeiten
- Informatikspezifische Methoden und Werkzeuge
- Anfangsunterricht und Projektunterricht
- Laborarbeit im Informatikunterricht
- Lehrpläne, Unterrichtsbeispiele
- Leistungsbewertung und Evaluation
- Erarbeitung von Lehr- und Lernmaterialien

Literaturhinweise

Rüdiger Baumann: Didaktik der Informatik

Peter Hubwieser: Didaktik der Informatik

S. Schubert, A. Schwill: Didaktik der Informatik

Weitere Informationen

Empfehlung:

Die Module Programmierung 1 und 2 und Einführung in die Theoretische Informatik sollten vor Beginn der Veranstaltung besucht worden sein

Je nach Thema ist die vorherige Teilnahme an weiteren korrespondierenden Veranstaltungen der Fachwissenschaft ratsam.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	3-7	jährlich, i.d.R. im SomSem	1 Semester	2 SWS + 15 Tage	7

Modulverantwortliche/r Prof. Dr. Verena Wolf

Dozent/inn/en OStR Stefan Strobel

Zulassungsvoraussetzungen Erfolgreiche Absolvierung des Orientierungspraktikums

Leistungskontrollen / Prüfungen

- Mündliche Prüfung im Begleitseminar
- Praktikumsbericht

Lehrveranstaltungen / SWS

- Semesterbegleitendes fachdidaktisches Schulpraktikum (15 Unterrichtstage an einer Schule, die dem angestrebten Lehramt entspricht)
- Begleitendes Seminar (2 SWS)

Arbeitsaufwand Schulpraktikum (120 Stunden) :

Präsenzzeit, Zeit für die Vor- und Nacharbeitung der Unterrichtshospitationen und eigener Unterrichtsversuche

Begleitseminar (90 Stunden):

30 h Präsenzzeit, 60 h für die Vor- und Nacharbeitung der Seminarsitzungen sowie das Erstellen des Praktikumsberichtes

120 h Präsenzstudium
+ 90 h Eigenstudium
= 210 h (= 7 ECTS)

Modulnote unbenotet

Sprache Deutsch

Lernziele / Kompetenzen

Die Studierenden

- kennen die aktuellen Themen der Schulinformatik gemäß den gültigen Lehrplänen
- können aus der Beobachtung und der Analyse von Informatikunterricht Hinweise und Erfahrungen für eigenes Planen und Handeln ableiten
- lernen unterschiedliche Methoden, Unterrichtsformen und Medien kennen
- werden mit Werkzeugen des Informatikunterrichtes vertraut
- erwerben erste eigene Erfahrungen bei der Planung und Umsetzung von Informatikunterricht
- lernen das Berufsbild des Informatiklehrers in der Praxis kennen und erhalten Hinweise auf ihre persönliche Eignung und Motivation
- können kooperativ im Team arbeiten und lernen

Inhalt

- Grundlegende Konzepte und Inhalte von Informatikunterricht
- Lehr- und Lernprozesse des Informatikunterrichtes
- Methoden, Arbeits- und Sozialformen des Informatikunterrichtes
- Strategien bei der didaktischen Reduktion und der Inhaltsauswahl
- Fragen der Organisation von Informatikunterricht, informatikspezifische Arbeitsformen
- Medien im Informatikunterricht
- Vorgehensweise bei der Aufbereitung und der unterrichtlichen Einbindung fachwissenschaftlicher Inhalte
- Informatikspezifische Methoden und Werkzeuge
- Lehrpläne, Unterrichtsbeispiele
- Erste eigene Unterrichtsversuche

Literaturhinweise

Die Literatur wird in der Veranstaltung bekannt gegeben.

Weitere Informationen

Empfehlung für LAB und LS1+2: Eine erfolgreiche Absolvierung der Pflichtmodule Programmierung 1 und 2, Systemarchitektur und Grundzüge der Theoretischen Informatik vor Besuch der Veranstaltungen ist ratsam.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	2-7	jährlich	1 Semester	2	3

Modulverantwortliche/r Prof. Dr. Verena Wolf

Dozent/inn/en StD Tim Lethen

Zulassungsvoraussetzungen –

Leistungskontrollen / Prüfungen

- regelmäßige Teilnahme am Seminar
- Vorbereiten und Halten von Seminarvorträgen
- Anfertigung schriftlicher Ausarbeitungen

Lehrveranstaltungen / SWS 2 SWS Seminar

Arbeitsaufwand 30 h Präsenzstudium + 60 h Eigenstudium = 90 h (= 3 ECTS)

Modulnote Seminarvorträge und schriftliche Ausarbeitungen

Sprache Deutsch

Lernziele / Kompetenzen

Die Studierenden ...

- können informatische Sachverhalte in verschiedenen Anwendungsbezügen und Sachzusammenhängen sowie gesellschaftliche Auswirkungen erfassen, bewerten und erklären
- verfügen über erste Erfahrungen, die Langlebigkeit und Übertragbarkeit der zentralen informatischen Fachkonzepte verständlich für alle Schülerinnen und Schüler zu erklären
- kennen die verschiedenen Sichtweisen der Informatik mit ihren spezifischen Zugängen zur Erkenntnisgewinnung, wie Konstruieren, Beweisen und empirische Methoden
- können fachdidaktische Konzepte und empirische Befunde informatikbezogener Lehr-Lernforschung und Diagnosewerkzeuge nutzen, um individuelle Denkwege und Vorstellungen von Schülerinnen und Schülern je nach ihren persönlichen Voraussetzungen, Vorerfahrungen und Fähigkeiten zu analysieren, Schülerinnen und Schüler für das Lernen von Informatik zu motivieren sowie individuelle Lernfortschritte zu fördern und zu bewerten
- verfügen über erste reflektierte Erfahrungen in der kompetenzorientierten Planung und Durchführung von Informatikunterricht und kennen die Grundlagen der Leistungsdiagnose und Leistungsbeurteilung im Fach
- kennen unterschiedliche außerschulische Förderangebote (Informatikwettbewerbe, Kurse etc.) und ermutigen ihre Schülerinnen und Schüler je nach ihren individuellen Fähigkeiten zur Teilnahme

Inhalt

- didaktische (Re-)Konstruktion fachlichen Wissens, insbesondere didaktische Reduktion (Beispiele)
- Kenntnis, Analyse und didaktische Aufbereitung von verschiedenen Kontexten zur Motivation aller Schülerinnen und Schüler
- Methoden, Techniken und Medien zur Erschließung informatischer Inhalte, so dass die visuelle, auditive und haptische Wahrnehmung angesprochen und die Regeln für leichte Sprache beachtet werden
- historische und aktuelle Unterrichtsansätze und typische Unterrichtsmethoden und -techniken der Informatik

Literaturhinweise

Wird in der Veranstaltung bekanntgegeben.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	2-7	jährlich	1 Semester	2	3

Modulverantwortliche/r Prof. Dr. Verena Wolf

Dozent/inn/en StD Tim Lethen

Zulassungsvoraussetzungen –

Leistungskontrollen / Prüfungen

- regelmäßige Teilnahme am Seminar
- Vorbereiten und Halten von Seminarvorträgen
- Anfertigung schriftlicher Ausarbeitungen

Lehrveranstaltungen / SWS 2 SWS Seminar

Arbeitsaufwand 30 h Präsenzstudium + 60 h Eigenstudium = 90 h (= 3 ECTS)

Modulnote Seminarvorträge und schriftliche Ausarbeitungen

Sprache Deutsch

Lernziele / Kompetenzen

Die Studierenden ...

- können Bezüge zwischen ihrem Fachwissen und der Schulinformatik herstellen, Unterrichtskonzepte und -medien auch für heterogene Lerngruppen fachlich gestalten, inhaltlich bewerten, neuere informatische Forschung in Überichtsdarstellungen verfolgen und neue Themen adressatengerecht in den Unterricht einbringen
- können in ersten Ansätzen die Darstellung und Erklärung von informatischen Unterrichtsinhalten methodisch, ggf. zielfähig und in Abstimmung mit anderen pädagogischen Fachkräften an die Bedürfnisse einer heterogenen Schülerschaft anpassen; sie verwenden dazu insbesondere eine reflektierte, konsistente Auswahl informatischer Fachbegriffe
- kennen Möglichkeiten zur Illustration von informatischen Prinzipien, welche die visuelle, auditive und haptische Wahrnehmung ansprechen und Regeln für leichte Sprache
- verfügen über ausreichende praktische Kompetenz für den Einsatz von schulrelevanter Hard- und Software, sie können insbesondere die Möglichkeiten, die sich durch den Einsatz von assistiven Technologien im Informatikunterricht eröffnen, einschätzen und bewerten
- sind in der Lage, Entwicklungen im Bereich Digitalisierung aus fachlicher und fachdidaktischer Sicht angemessen zu rezipieren sowie Möglichkeiten und Grenzen der Digitalisierung kritisch zu reflektieren. Sie können die daraus gewonnenen Erkenntnisse in fachdidaktischen Kontexten nutzen sowie in die Weiterentwicklung unterrichtlicher und curricularer Konzepte einbringen. Sie sind sensibilisiert für die Chancen digitaler Lernmedien hinsichtlich Barrierefreiheit und nutzen digitale Medien auch zur Differenzierung und individuellen Förderung im Unterricht

Inhalt

- didaktische (Re-)Konstruktion fachlichen Wissens, insbesondere didaktische Reduktion (Beispiele)
- Kenntnis, Analyse und didaktische Aufbereitung von verschiedenen Kontexten zur Motivation aller Schülerinnen und Schüler
- Methoden, Techniken und Medien zur Erschließung informatischer Inhalte, so dass die visuelle, auditive und haptische Wahrnehmung angesprochen und die Regeln für leichte Sprache beachtet werden
- historische und aktuelle Unterrichtsansätze und typische Unterrichtsmethoden und -techniken der Informatik

Literaturhinweise

Wird in der Veranstaltung bekanntgegeben.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	3-5	jedes Wintersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Raimund Seidel

Dozent/inn/en Prof. Dr. Raimund Seidel
Prof. Dr. Bernd Finkbeiner
Prof. Dr. Kurt Mehlhorn
Prof. Dr. Markus Bläser

Zulassungsvoraussetzungen *Programmierung 1 und 2 und Mathematik für Informatiker 1 und 2 oder vergleichbare Veranstaltungen der Mathematik sind empfohlen.*

Leistungskontrollen / Prüfungen Erfolgreiche Bearbeitung der Übungsaufgaben berechtigt zur Klausurteilnahme.

Lehrveranstaltungen / SWS 4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS

Arbeitsaufwand 90 h Präsenzstudium
+ 180 h Eigenstudium
= 270 h (= 9 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Englisch

Lernziele / Kompetenzen

Die Studierenden kennen verschiedene Rechenmodelle und ihre relativen Stärken und Mächtigkeiten.

Sie können für ausgewählte Probleme zeigen, ob diese in bestimmten Rechenmodellen lösbar sind oder nicht.

Sie verstehen den formalen Begriff der Berechenbarkeit wie auch der Nicht-Berechenbarkeit.

Sie können Probleme aufeinander reduzieren.

Sie sind vertraut mit den Grundzügen der Ressourcenbeschränkung (Zeit, Platz) für Berechnungen und der sich daraus ergebenden Komplexitätstheorie.

Inhalt

Die Sprachen der Chomsky Hierarchie und ihre verschiedenen Definitionen über Grammatiken und Automaten; Abschlusseigenschaften; Klassifikation von bestimmten Sprachen („Pumping lemmas“);

Determinismus und Nicht-Determinismus;

Turing Maschinen und äquivalente Modelle von allgemeiner Berechenbarkeit (z.B. μ -rekursive Funktionen, Random Access Machines) Reduzierbarkeit, Entscheidbarkeit, Nicht-Entscheidbarkeit;

Die Komplexitätsmaße Zeit und Platz; die Komplexitätsklassen P und NP;

Grundzüge der Theorie der NP-Vollständigkeit

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Introduction to Theoretical Computer Science*.

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

2-6

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	1-3	jedes Wintersemester	1 Semester	4	6

Modulverantwortliche/r Prof. Dr. Verena Wolf

Dozent/inn/en StD Tim Lethen

Zulassungsvoraussetzungen Keine

Leistungskontrollen / Prüfungen

- Teilnahme an den Übungen und erfolgreiche Bearbeitung der wöchentlichen Übungsaufgaben
- Abschluss-/Nachklausur
- Die Nachklausur findet i.d.R. innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.
- Statt einer schriftlichen Abschlussklausur und Nachklausur kann eine mündliche Prüfung erfolgen (Entscheidung des Dozenten)

Lehrveranstaltungen / SWS

2 SWS Vorlesung
+ 2 SWS Übung
= 4 SWS

Arbeitsaufwand

60 h Präsenzstudium
+ 120 h Eigenstudium
= 180 h (= 6 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Deutsch

Lernziele / Kompetenzen

- Erarbeitung von mathematischem Grundlagenwissen, das im Rahmen eines Informatiklehramtsstudiums benötigt wird
- Fähigkeit zur Formalisierung und Abstraktion
- Befähigung zur Aneignung weiteren mathematischen Wissens mit Hilfe von Lehrbüchern

Inhalt

1. Einführung in die Mengenlehre
 1. Mengenoperationen
 2. Paare und Tupel
 3. Relationen
 4. Abbildungen
 5. Beispiel: Graphen und Isomorphie
 6. Unendlichkeit: Abzählbarkeit und Aufzählbarkeit
 7. Mengentheoretische Antinomien
2. Einführung in die Logik
 1. Aussagenlogik
 1. Syntax
 2. Beweise nach Gentzen und Hilbert
 3. Intuitionistische und substrukturelle Logiken
 4. Wahrheitswerttabellen und Boole'sche Algebren
 2. Prädikatenlogik erster Stufe
 1. Syntax

- 2. Semantik
- 3. Modallogik
 - 1. Syntax
 - 2. Kripke-Semantik
- 3. Natürliche Zahlen und Modular-Arithmetik
 - 1. Peano-Axiome und Ordinalzahlen
 - 2. Vollständige und strukturelle Induktion
 - 3. Widerspruchsbeweise
 - 4. Modular-Arithmetik
- 4. Eindimensionale Analysis
 - 1. Reelle Zahlen, Folgen und Reihen
 - 2. Differentialrechnung
 - 3. Integralrechnung

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	2-4	jedes Sommersemester	1 Semester	2	3

Modulverantwortliche/r Prof. Dr. Verena Wolf

Dozent/inn/en StD Tim Lethen

Zulassungsvoraussetzungen Mathematik für Informatik-Lehramt 1 (empfohlen)

Leistungskontrollen / Prüfungen

- Teilnahme an den Präsenzübungen
- Bestehen der Abschlussklausur oder der Nachklausur
- Die Nachklausur findet i.d.R. innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.
- Statt einer schriftlichen Abschlussklausur und Nachklausur kann eine mündliche Prüfung erfolgen.

Lehrveranstaltungen / SWS

2 SWS Vorlesung
+ 1 SWS Übung
= 3 SWS

Arbeitsaufwand

30 h Präsenzstudium
+ 60 h Eigenstudium
= 90 h (= 3 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Deutsch

Lernziele / Kompetenzen

- Erarbeitung von mathematischem Grundlagenwissen, das im Rahmen eines Informatiklehramtsstudiums benötigt wird
- Fähigkeit zur Formalisierung und Abstraktion
- Befähigung zur Aneignung weiteren mathematischen Wissens mit Hilfe von Lehrbüchern

Inhalt

LINEARE ALGEBRA UND WAHRSCHEINLICHKEITSRECHNUNG

C. LINEARE ALGEBRA (8)

- 21. Der Vektorraum $\mathbb{R}^n$ (2)
- 22. Matrizen und lineare Gleichungssysteme (4)
 - Matrizenkalkül
 - Lösbarkeit linearer Gleichungssysteme
 - Invertierbare Matrizen \newline - Determinanten
 - Spezialfälle quadratischer Matrizen
 - Hauptachsentransformation: Eigenwerte und Eigenvektoren
- 23. Grenzwerte und Stetigkeit im $\mathbb{R}^n$ (2)

D. WAHRSCHEINLICHKEITSRECHNUNG (7)

- 24. Kombinatorik (2)
- 24. Wahrscheinlichkeit(räume), bedingte Wahrscheinlichkeiten (2)
- 25. Zufallsvariablen, Erwartung, Varianz (2)
- 26. Zentraler Grenzwertsatz (1)

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	1-3	jedes Wintersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Gert Smolka

Dozent/inn/en Prof. Dr. Gert Smolka
Prof. Dr.-Ing. Holger Hermanns
Prof. Bernd Finkbeiner, Ph.D

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

- zwei Klausuren (Mitte und Ende der Vorlesungszeit)
- Die Note wird aus den Klausuren gemittelt und kann durch Leistungen in den Übungen verbessert werden.
- Eine Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.

Lehrveranstaltungen / SWS 4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS

Arbeitsaufwand 90 h Präsenzstudium
+ 180 h Eigenstudium
= 270 h (= 9 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Deutsch und Englisch

Lernziele / Kompetenzen

- höherstufige, getypte funktionale Programmierung anwenden können
- Verständnis rekursiver Datenstrukturen und Algorithmen, Zusammenhänge mit Mengenlehre
- Korrektheit beweisen und Laufzeit abschätzen
- Typabstraktion und Modularisierung verstehen
- Struktur von Programmiersprachen verstehen
- einfache Programmiersprachen formal beschreiben können
- einfache Programmiersprachen implementieren können
- anwendungsnahe Rechenmodelle mit maschinennahen Rechenmodellen realisieren können
- Praktische Programmiererfahrung, Routine im Umgang mit Interpretern und Übersetzern

Inhalt

- Funktionale Programmierung
- Algorithmen und Datenstrukturen (Listen, Bäume, Graphen; Korrektheitsbeweise; asymptotische Laufzeit)
- Typabstraktion und Module
- Programmieren mit Ausnahmen
- Datenstrukturen mit Zustand
- Struktur von Programmiersprachen (konkrete und abstrakte Syntax, statische und dynamische Syntax)
- Realisierung von Programmiersprachen (Interpreter, virtuelle Maschinen, Übersetzer)

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Programming 1*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	2-4	jedes Sommersemester	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Sebastian Hack

Dozent/inn/en Prof. Dr. Sebastian Hack
Prof. Dr. Jörg Hoffmann

Zulassungsvoraussetzungen *Programmierung 1* und *Mathematik für Informatiker 1* und Mathematikveranstaltungen im Studiensemester oder vergleichbare Kenntnisse aus sonstigen Mathematikveranstaltungen (empfohlen)

Leistungskontrollen / Prüfungen Prüfungsleistungen werden in zwei Teilen erbracht, die zu gleichen Teilen in die Endnote eingehen. Um die Gesamtveranstaltung zu bestehen, muss jeder Teil einzeln bestanden werden.

Im **Praktikumsteil** müssen die Studierenden eine Reihe von Programmieraufgaben selbstständig implementieren. Diese Programmieraufgaben ermöglichen das Einüben der Sprachkonzepte und führen außerdem komplexere Algorithmen und Datenstrukturen ein. Automatische Tests prüfen die Qualität der Implementierungen. Die Note des Praktikumsteils wird maßgeblich durch die Testergebnisse bestimmt.

Im **Vorlesungsteil** müssen die Studierenden Klausuren absolvieren und Übungsaufgaben bearbeiten. Die Aufgaben vertiefen dabei den Stoff der Vorlesung. Die Zulassung zu der Klausur hängt von der erfolgreichen Bearbeitung der Übungsaufgaben ab.

Im Praktikumsteil kann eine Nachaufgabe angeboten werden

Lehrveranstaltungen / SWS 4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS

Arbeitsaufwand 90 h Präsenzstudium
+ 180 h Eigenstudium
= 270 h (= 9 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben

Sprache Deutsch und Englisch

Lernziele / Kompetenzen

Die Studierenden lernen die Grundprinzipien der imperativen /objektorientierten Programmierung kennen. Dabei wird primär Java als Programmiersprache verwendet.

In dieser Vorlesung lernen sie:

- wie Rechner Programme ausführen
- Die Grundlagen imperativer und objektorientierter Sprachen
- kleinere, wohlstrukturierte Programme in C zu schreiben
- mittelgroße objektorientierte Systeme in Java zu implementieren und zu testen
- sich in wenigen Tagen eine neue imperative/objektorientierte Sprache anzueignen, um sich in ein bestehendes Projekt einzuarbeiten

Inhalt

- Imperatives Programmieren
- Objekte und Klassen
- Klassendefinitionen
- Objektinteraktion
- Objektsammlungen
- Objekte nutzen und testen
- Vererbung
- Dynamische Bindung
- Fehlerbehandlung
- Klassendesign und Modularität
- Systemnahe Programmierung

sowie spezifische Vorlesungen für die Programmieraufgaben.

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Programming 2*.

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

2-6

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

3-5

lecture free time after SS

7 weeks

BLOCK**9****Modulverantwortliche/r****Dozent/inn/en**

Zulassungsvoraussetzungen Participation in the Software Engineering Lab requires extensive programming skills as taught in the courses *Programming 1* and *Programming 2*. Passing *Programming 2* is recommended before enrolling in this course.

Students are required to bring their own laptops.

Leistungskontrollen / Prüfungen The goal of the Software Engineering Lab is to develop a non-trivial software system in a team effort. In this course, a number of documents (design models, documentation, implementation plan, etc.) and artifacts (source code, tests, etc.) need to be developed and submitted. Correctness, completeness, quality, and timely submission of all documents and artifacts are major grading criteria.

The Software Engineering Lab consists of two phases: exercise phase and group phase.

In the *exercise phase*, participants will complete an entry exam, covering current topics from the lecture. Only participants that have passed the exercise phase will be admitted to the group phase.

In the *group phase*, participants will first design and then implement and test a substantial software system in a team effort, and submit both their design and their implementation (including tests) for evaluation. All documents (design models, documentation, implementation plan, etc.) and artifacts (source code, tests, etc.) of the group phase will be evaluated based on the principles and quality criteria conveyed in the lectures. To pass the group phase, students must pass both the design submission and the implementation submission, and prove individually their substantial contribution to the group project.

More details on the exams will be announced at the beginning of the course.

Lehrveranstaltungen / SWS Daily exercises and lectures (first few weeks)
Daily project work with tutoring

Arbeitsaufwand 35 h of lectures and exercises
+ 235 h project work
= 270 h (= 9 ECTS)

Modulnote ungraded

Sprache English

Lernziele / Kompetenzen

Participants acquire the ability to solve complex software development problems individually and in teams.

Participants are aware of common problems and pitfalls of software development and know how to address them.

Participants are able to accomplish and coordinate software development tasks based on a set of given requirements. For this purpose, they are able to select proper methods and techniques to minimize risks and maximize software quality.

Participants know about foundations and principles of software design, including cohesion, coupling, modularity, encapsulation, abstraction, and information hiding. They are acquainted with a whole array of design patterns, knowing their aim and individual strengths and weaknesses. They are able to apply design patterns beneficially and to judge and improve the quality of software designs.

Participants master fundamental techniques and tools for software testing, debugging, and version control.

Inhalt

- Software design
- Software testing
- Team work
- Debugging

Literaturhinweise

- Software Engineering. I. Sommerville, Addison-Wesley, 2004.
- Software Engineering: A Practitioner's Approach. R. Pressman, McGraw Hill Text, 2001.
- Using UML: Software Engineering with Objects and Components. P. Stevens, et al., Addison-Wesley, 1999.
- UML Distilled. M. Fowler, et al., Addison-Wesley, 2000.
- Objects, Components and Frameworks with UML, D. D'Souza, et al., Addison-Wesley, 1999.
- Designing Object-Oriented Software. R. Wirfs-Brock, et al., Prentice Hall, 1990.
- Design Patterns. Elements of Reusable Object-Oriented Software. E. Gamma, et al., Addison-Wesley, 1995.
- Head First Design Patterns. E. Freeman, et al. O'Reilly, 2004.
- Software Architecture: Perspectives on an Emerging Discipline. M. Shaw, et al., Prentice-Hall, 1996.
- Refactoring: Improving the Design of Existing Code. M. Fowler, et al., Addison-Wesley, 1999.
- Software Testing and Analysis: Process, Principles and Techniques. M. Pezze, Wiley. 2007.

Weitere Informationen

This module is identical in content to the German-language module *Softwarepraktikum*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	3-5	jedes Sommersemester	1 Semester	4	6

Modulverantwortliche/r Prof. Dr. Jan Reineke

Dozent/inn/en Prof. Dr. Jan Reineke

Zulassungsvoraussetzungen Empfohlene Vorkenntnisse:
 Programmierung 1 und Mathematik für Lehramt Informatik 1 oder vergleichbare Veranstaltungen der Mathematik

Leistungskontrollen / Prüfungen Studienleistungen: die Vorlesungen hören, nachbearbeiten und gegebenenfalls verstehen; die Übungen allein oder in Gruppen bearbeiten.
 Prüfungsleistungen: erfolgreiche Bearbeitung der Minutests oder der Übungen berechtigt zur Teilnahme an den Klausuren
 Die Nachklausur findet i.d.R. innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.

Lehrveranstaltungen / SWS Vorlesung: 3 SWS (ca. 10 Studierende, teilweise zusammen mit Hörern des Moduls Systemarchitektur)
 Übung: 1 SWS
 Übungsgruppen mit bis zu 20 Studierenden

Arbeitsaufwand 60 h Präsenzstudium
 + 120 h Eigenstudium
 = 180 h (= 9 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Deutsch oder Englisch

Lernziele / Kompetenzen

Die Studierenden sollen elementare Prinzipien des Aufbaus moderner Prozessoren und Betriebssysteme lernen. Auf dieser Basis sollen sie in die Lage versetzt werden solche Systeme zu erweitern und die Korrektheit ihrer Erweiterungen zu beweisen.

Inhalt

1. Hardware
 - a. Boolesche Algebra und Schaltkreise
 - b. Kodierung von Zahlen und arithmetische Schaltkreise
 - c. Einfacher, sequentieller Prozessor
 - d. Leistungsoptimierungen wie Caches und Pipelining
2. Betriebssystemgrundlagen
 - a. Anforderungen an Betriebssysteme
 - b. Grundlegende Mechanismen zur Realisierung von Betriebssystemen

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Empfohlene Vorkenntnisse:

Programmierung 1 und Mathematik für Lehramt Informatik 1 oder vergleichbare Veranstaltungen der Mathematik

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	3-7	jährlich	1 Semester	2 SWS+4 Wochen	9

Modulverantwortliche/r Prof. Dr. Verena Wolf

Dozent/inn/en OStR Stefan Strobel

Zulassungsvoraussetzungen Elementare fachdidaktische schulpraktische Studien

Leistungskontrollen / Prüfungen

- Erfolgreiche Teilnahme am Begleitseminar
- Praktikumsbericht

Lehrveranstaltungen / SWS

- Vierwöchiges Praktikum an einer Schule, die dem angestrebten Lehramt entspricht
- Vor- und nachbereitendes Seminar (2 SWS)

Arbeitsaufwand Schulpraktikum (180 Stunden) :

Präsenzzeit, Zeit für die Vor- und Nacharbeitung der Unterrichtshospitationen und eigener Unterrichtsversuche

Begleitseminar (90 Stunden):

30 h Präsenzzeit, 60 h für die Vor- und Nacharbeitung der Seminarsitzungen sowie das Erstellen des Praktikumsberichtes

210 h Präsenzstudium
+ 60 h Eigenstudium
= 270 h (= 9 ECTS)

Modulnote benotet

Sprache Deutsch

Lernziele / Kompetenzen

Die Studierenden erwerben Kompetenzen

- in der sach- und fachgerechten Planung und Durchführung von Informatikunterricht
- in der Reflektion und Evaluation von Informatikunterricht
- in der Wahl geeigneter fachlicher Inhalte und fachspezifischer Methoden, Arbeits- und Kommunikationsformen
- in der Gestaltung von Lehr- und Lernprozessen
- in der Wertung und Beurteilung von Schülerleistungen
- beim Herstellen von Lernbereitschaft und Motivation bei den Schülerinnen und Schülern.

Sie gewinnen weitere Erfahrungen in der Anwendung fachlichen und fachdidaktischen Wissens in der schulischen Praxis. Persönliche Erfahrungen im Team und bei den ersten Unterrichtsversuchen helfen ihnen bei der Entwicklung einer eigenen Lehrerpersönlichkeit und bei der Berufsorientierung.

Inhalt

- Methoden, Arbeits- und Sozialformen des Informatikunterrichtes
- Strategien bei der didaktischen Reduktion und der Inhaltsauswahl
- Vorgehensweisen bei der Unterrichtsplanung, der Lernzielbeschreibung und deren Dokumentation
- Fragen der Organisation von Informatikunterricht, informatikspezifische Arbeitsformen
- Planung, Durchführung und Evaluation eigenen Informatikunterrichtes

Literaturhinweise

Die Literatur wird in der Veranstaltung bekannt gegeben.

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

2-4

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

1-5

Modulverantwortliche/r

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Modulbereich 2

Wahlpflichtbereich 1

English for Physical Sciences

Studiensem.

Regelst.sem.

Turnus

Dauer

SWS

ECTS

3-5

Sommersemester

2

2

Modulverantwortliche/r Dr. Peter Tischer

Dozent/inn/en

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Arbeitsaufwand

Modulnote

Sprache

Lernziele / Kompetenzen

Inhalt

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	3-5	jedes Wintersemester	1 Semester	2	2

Modulverantwortliche/r Studiendekan der Fakultät Mathematik und Informatik
Studienbeauftragter der Informatik

Dozent/inn/en Dozent/inn/en der Fachrichtung

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen Nachweis über das inhaltliche Verständnis von mindestens drei Vorträgen, z.B. durch schriftliche Ausarbeitung oder Test.

Lehrveranstaltungen / SWS 2 SWS Vorlesung

Arbeitsaufwand 30 h Präsenzstudium
+ 30 h Eigenstudium
= 60 h (= 2 ECTS)

Modulnote Das Modul ist insgesamt bestanden, wenn die Prüfungsleistung bestanden wurde (unbenotet).

Sprache Deutsch / English

Lernziele / Kompetenzen

Frühzeitige Motivierung und Überblick über die zentralen wissenschaftlichen Fragestellungen der Informatik, sowie über die Kompetenzen der Saarbrücker Informatik.

Inhalt

Vorträge durch wöchentlich wechselnde Dozent/inn/en bieten einen Querschnitt durch die Forschungsthemen der Saarbrücker Informatik. Die Themen spannen einen attraktiven Bogen von aktuellster Forschung zu anspruchsvollen Problemen der industriellen Praxis.

Literaturhinweise

Material wird passend zu den zu den einzelnen Vorträgen bereitgestellt.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Perspectives in Computer Science*.

Modulbereich 3

Wahlpflichtbereich 2

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	5-7	every winter semester	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Jörg Hoffmann
Prof. Dr. Jens Dittrich
Prof. Dr. Bernt Schiele
Prof. Dr. Vera Demberg

Dozent/inn/en Prof. Dr. Jörg Hoffmann
Prof. Dr. Jens Dittrich
Prof. Dr. Bernt Schiele
Prof. Dr. Vera Demberg

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen Weekly assignments,
Exam (qualification for exam depends on performance in assignments)

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Based on exam. The exact modalities are specified by the lecturers.

Sprache English

Lernziele / Kompetenzen

Overview of challenges and methods in Data Science and AI. Basic knowledge of key concepts and algorithms.

Inhalt

Introduction to history and concepts of Data Science and AI

- Machine Learning (supervised, unsupervised, reinforcement, neural networks)
- (adversarial) Search, Planning
- Reasoning
- Modeling and Simulation
- Data Management, Big Data Engineering, and Analytics

The methods will be covered in the context of applications, such as Game Playing, Computer Vision, Autonomous Driving, Language Processing, Social Networks.

The exercises will cover methodological, algorithmic, as well as practical aspects. Where basic programming or scripting skills are required, the lecture and exercises will introduce these skills.

Literaturhinweise

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	5-7	every winter semester	1 semester	4	6

Modulverantwortliche/r Prof. Dr. Jilles Vreeken
Prof. Dr. Isabel Valera

Dozent/inn/en Prof. Dr. Jilles Vreeken
Prof. Dr. Isabel Valera

Zulassungsvoraussetzungen The lecture assumes basic knowledge in statistics, linear algebra, and programming. It is advisable to have successfully completed *Mathematics for Computer Scientists 2* and *Statistics Lab*. The exercises use the programming language R. We will give a basic introduction to R in the first tutorial. In addition, for preparation the following materials are useful: *R for Beginners* by Emmanuel Paradis (especially chapters 1, 2, 3 and 6) and *An introduction to R* (Venables/Smith).

Leistungskontrollen / Prüfungen Prerequisite for admission to the examination is a cumulative 50% of the points of the theoretical and a cumulative 50% of the points of the practical tasks on the exercise sheets. Depending on the number of participants, the examinations are either written or oral. The final modality will be announced in the first two weeks of the lecture.

Lehrveranstaltungen / SWS 2 h lectures
+ 2 h tutorial
= 4 h (weekly)

Arbeitsaufwand 60 h of classes
+ 120 h private study
= 180 h (= 6 ECTS)

Modulnote Will be determined from performance in exams.

Sprache English

Lernziele / Kompetenzen

In this course we will discuss the foundations – the elements – of machine learning. In particular, we will focus on the ability of, given a data set, to choose an appropriate method for analyzing it, to select the appropriate parameters for the model generated by that method and to assess the quality of the resulting model. Both theoretical and practical aspects will be covered. What we cover will be relevant for computer scientists in general as well as for other scientists involved in data analysis and modeling.

Inhalt

The lecture covers basic machine learning methods, in particular the following contents:

- Introduction to statistical learning
- Overview over Supervised Learning
- Linear Regression
- Linear Classification
- Splines
- Model selection and estimation of the test errors
- Maximum-Likelihood Methods
- Additive Models
- Decision trees

- Boosting
- Dimensionality reduction
- Unsupervised learning
- Clustering
- Visualization

Literaturhinweise

The course broadly follows the book *An Introduction to Statistical Learning with Applications in R*, Springer (2013). In some cases, the course receives additional material from the book *The Elements of Statistical Learning*, Springer (second edition, 2009). The first book is the introductory text, the second covers more advanced topics. Both books are available as free PDFs. Any change of, or additional material will be announced before the start of the course on the course webpage.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	5-7	every winter semester	1 semester	6	9

Modulverantwortliche/r Dr. Ben Stock

Dozent/inn/en Dr. Ben Stock

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen Students need to solve the exercise during the semester to be allowed to take the exam.

Lehrveranstaltungen / SWS 4 SWS Lecture
+ 2 SWS Tutorials
= 6 SWS

Arbeitsaufwand 90 h of classes
+ 180 h of private study
= 270 h (= 9 ECTS)

Modulnote By default, the final grade is calculated through the exam only. This mode can be changed by the lecturer and such changes will be announced at the beginning of the term.

Sprache English

Lernziele / Kompetenzen

The students know the legal foundations of information security in Germany. In addition, they know the basic building blocks of modern cryptography, network security as well as privacy. Special emphasis is on network security, such that students know relevant protocols for secure communication and can utilize them.

To apply the learned theoretical knowledge, the students will also learn Python to apply the concepts in practical tasks.

Inhalt

- Foundations of the Strafgesetzbuch w.r.t. to information security
- Basic understanding of symmetric and asymmetric cryptographic protocol and their usage scenarios
- Basic understanding of hash functions and important properties of hash functions
- Network foundations of all layer (according to the TCP/IP model)
- Security protocols for each network layer
- Foundations of privacy and anonymity
- Basics of Web security
- Parallel to the security topics, we will also introduce the Python programming language

Literaturhinweise

The literature is english and will be announced at the beginning of the lecture.

Weitere Informationen

Programming tasks in Python. Pen&paper exercises in groups (and tutorials).

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	5-7	jedes Wintersemester	1 Semester	4	6

Modulverantwortliche/r Prof. Dr. Raimund Seidel

Dozent/inn/en Prof. Dr. Raimund Seidel
Prof. Dr. Kurt Mehlhorn
Prof. Dr. Markus Bläser

Zulassungsvoraussetzungen *Programmierung 1 und 2, und Mathematik für Informatiker 1 und 2 oder vergleichbare Veranstaltungen der Mathematik sind empfohlen.*

Leistungskontrollen / Prüfungen Erfolgreiche Bearbeitung der Übungsblätter berechtigt zur Klausurteilnahme.

Lehrveranstaltungen / SWS 2 SWS Vorlesung
+ 2 SWS Übung
= 4 SWS

Arbeitsaufwand 60 h Präsenzstudium
+ 120 h Eigenstudium
= 180 h (= 6 ECTS)

Modulnote Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Englisch

Lernziele / Kompetenzen

Die Studierenden lernen die wichtigsten Methoden des Entwurfs von Algorithmen und Datenstrukturen kennen: Teile-und-Herrsche, Dynamische Programmierung, inkrementelle Konstruktion, „Greedy“, Dezimierung, Hierarchisierung, Randomisierung. Sie lernen Algorithmen und Datenstrukturen bzgl. Zeit- und Platzverbrauch für das übliche RAM Maschinenmodell zu analysieren und auf Basis dieser Analysen zu vergleichen. Sie lernen verschiedene Arten der Analyse (schlechtester Fall, amortisiert, erwartet) einzusetzen.

Die Studierenden lernen wichtige effiziente Datenstrukturen und Algorithmen kennen. Sie sollen die Fähigkeit erwerben, vorhandene Methoden durch theoretische Analysen und Abwägungen für ihre Verwendbarkeit in tatsächlich auftretenden Szenarien zu prüfen. Ferner sollen die Studierenden die Fähigkeit trainieren, Algorithmen und Datenstrukturen unter dem Aspekt von Performanzgarantien zu entwickeln oder anzupassen

Inhalt

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Fundamentals of Data Structures and Algorithms*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	5-7	jedes Sommersemester	1 Semester	4	6

Modulverantwortliche/r Prof. Dr.-Ing. Holger Hermanns

Dozent/inn/en Prof. Dr.-Ing. Holger Hermanns
Prof. Bernd Finkbeiner, Ph.D
Prof. Dr. Verena Wolf

Zulassungsvoraussetzungen *Programmierung 1 und 2, Softwarepraktikum, und Grundzüge der Theoretischen Informatik (empfohlen)*

Leistungskontrollen / Prüfungen Zwei Klausuren (Mitte und Ende der Vorlesungszeit), praktisches Projekt.
Eine Nachklausur für die Zwischenklausur findet vor der Hauptklausur statt, eine Nachklausur für die Hauptklausur findet in den letzten Wochen vor Vorlesungsbeginn des Folgesemesters statt.

Lehrveranstaltungen / SWS 4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS

Arbeitsaufwand 50 h Präsenzstudium
+ 130 h Eigenstudium
= 180 h (= 6 ECTS)

Modulnote Wird aus Leistungen in den Klausuren, sowie den Prüfungsvorleistungen ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Sprache Englisch

Lernziele / Kompetenzen

Die Teilnehmer lernen die Nebenläufigkeit von Berechnungen als ein weitreichendes, grundlegendes Prinzip in der Theorie und Anwendung der modernen Informatik kennen. Durch die Untersuchung und Verwendung unterschiedlicher formaler Modelle gewinnen die Teilnehmer ein vertieftes Verständnis von Nebenläufigkeit. Dabei lernen die Teilnehmer wichtige formale Konzepte der Informatik korrekt anzuwenden. Das im ersten Teil der Veranstaltung erworbene theoretische Wissen wird in der zweiten Hälfte in der (Programmier-)Praxis angewendet. Dabei lernen die Teilnehmer Verwendung der Programmierparadigmen „Shared Memory“ und „Message Passing“ zuerst gemeinsam in der Programmiersprache *pseuCo*, bevor sie dann diese Fähigkeiten auf Java und teilweise Rust übertragen. Außerdem lernen die Teilnehmer verschiedene Phänomene des nebenläufigen Programmierens in den formalen Modellen zu beschreiben und mit deren Hilfe konkrete Lösungen für die Praxis abzuleiten. Des Weiteren untersuchen die Teilnehmer in der Praxis existierende Konzepte auf ihre Verlässlichkeit hin. Ein spezifischer Aspekt dieser beruflichen Praxis ist das taktisch adäquate Reagieren auf Problemstellungen der Nebenläufigkeit unter engen Zeitvorgaben.

Inhalt

Nebenläufigkeit als Konzept

- Potentieller Parallelismus
- Tatsächlicher Parallelismus
- Konzeptioneller Parallelismus

Nebenläufigkeit in der Praxis

- Objektorientierung

- Betriebssysteme
- Multi-core Prozessoren, Coprozessoren
- Programmierte Parallelität
- Verteilte Systeme (Client-Server, Peer-to-Peer, Datenbanken, Internet)

Die Schwierigkeit von Nebenläufigkeit

- Ressourcenkonflikte
- Fairness
- Gegenseitiger Ausschluss
- Verklemmung (Deadlock)
- gegenseitige Blockaden (Livelock)
- Verhungern (Starvation)

Grundlagen der Nebenläufigkeit

- Sequentielle vs. Nebenläufige Prozesse
- Zustände, Ereignisse und Transitionen
- Transitionssysteme
- Beobachtbares Verhalten
- Determinismus vs. Nicht-Determinismus
- Algebren und Operatoren

CCS: Der Kalkül kommunizierender Prozesse

- Konstruktion von Prozessen: Sequenz, Auswahl, Rekursion
- Nebenläufigkeit und Interaktion
- Strukturelle operationelle Semantik
- Gleichheit von Beobachtungen
- Implementierungsrelationen
- CCS mit Datentransfer

Programmieren von Nebenläufigkeit

- pseuCo
- Message Passing in pseuCo und Go
- Shared Memory in pseuCo und Java
- Monitore und Semaphoren
- Shared Objects und Threads in Java
- Shared Objects und Threads als Transitionssysteme

Programmier- und Analyseunterstützung

- Erkennung von Verklemmungen
- Zusicherung von Sicherheit und Lebendigkeit
- Model-Basiertes Design von Nebenläufigkeit
- Software Architekturen für Nebenläufigkeit

Literaturhinweise

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Weitere Informationen

Dieses Modul ist inhaltsgleich mit dem englischsprachigen Modul *Concurrent Programming*.

Modulbereich 4

Wahlpflichtbereich 3

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Kurt Mehlhorn

Dozent/inn/en Prof. Dr. Raimund Seidel
Prof. Dr. Kurt Mehlhorn

Zulassungsvoraussetzungen For graduate students: C, C++, Java

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Passing the midterm and the final exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

The students know standard algorithms for typical problems in the area's graphs, computational geometry, strings and optimization. Furthermore, they master a number of methods and data-structures to develop efficient algorithms and analyze their running times.

Inhalt

- graph algorithms (shortest path, minimum spanning trees, maximal flows, matchings, etc.)
- computational geometry (convex hull, Delaunay triangulation, Voronoi diagram, intersection of line segments, etc.)
- strings (pattern matching, suffix trees, etc.)
- generic methods of optimization (tabu search, simulated annealing, genetic algorithms, linear programming, branch-and-bound, dynamic programming, approximation algorithms, etc.)
- data-structures (Fibonacci heaps, radix heaps, hashing, randomized search trees, segment trees, etc.)
- methods for analyzing algorithms (amortized analysis, average-case analysis, potential methods, etc.)

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Jörg Hoffmann

Dozent/inn/en Prof. Dr. Jörg Hoffmann

Zulassungsvoraussetzungen *Programming 1, Programming 2, Fundamentals of Data Structures and Algorithms, and Elements of Machine Learning* or other courses in machine learning are recommended.

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Solving of weekly assignments
- Passing the final written exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from the performance in exams. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

Knowledge about basic methods in Artificial Intelligence

Inhalt

Search:

- Uninformed- and informed search procedures
- Monte-Carlo tree search

Planning:

- Formalism and complexity
- Critical-path heuristics
- Delete relaxation heuristics
- Abstraction heuristics

Markov decision processes:

- Discounted reward and expected cost
- Value iteration
- Informed search
- Reinforcement learning

Games:

- Adversarial search
- Learning from self-play

Literaturhinweise

Russel & Norvig Artificial Intelligence: A Modern Approach;
further reading will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Christoph Weidenbach

Dozent/inn/en Prof. Dr. Christoph Weidenbach

Zulassungsvoraussetzungen *Introduction to Computational Logic*

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Weekly assignments
- Practical work with systems
- Passing the final and mid-term exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

The goal of this course is to provide familiarity with logics, calculi, implementation techniques, and systems providing automated reasoning.

Inhalt

Propositional Logic – CDCL, Superposition - Watched Literals
First-Order Logic without Equality – (Ordered) Resolution,
Equations with Variables – Completion, Termination
First-Order Logic with Equality – Superposition (SUP) - Indexing

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Sebastian Hack

Dozent/inn/en Prof. Dr. Sebastian Hack

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Written exam at the end of the course, theoretical exercises, and compiler-laboratory project.
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS

4 h lectures
 + 2 h tutorial
 = 6 h (weekly)

Arbeitsaufwand

90 h of classes
 + 180 h private study
 = 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

The students learn, how a source program is lexically, syntactically, and semantically analyzed, and how they are translated into semantically equivalent machine programs. They learn how to increase the efficiency by semantics-preserving transformations. They understand the automata-theoretic foundations of these tasks and learn, how to use the corresponding tools.

Inhalt

Lexical, syntactic, semantic analysis of source programs, code generation for abstract and real machines, efficiency-improving program transformations, foundations of program analysis.

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Markus Bläser

Dozent/inn/en Prof. Dr. Raimund Seidel
Prof. Dr. Markus Bläser

Zulassungsvoraussetzungen undergraduate course on theory of computation (e.g. *Grundzüge der Theoretischen Informatik*) is highly recommend.

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- assignments
- exams (written or oral)

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be calculated from the results in the assignments and/or exams, as announced by the Lecturer at the beginning of the course

Sprache English

Lernziele / Kompetenzen

The aim of this lecture is to learn important concepts and methods of computational complexity theory. The student shall be enabled to understand recent topics and results in computational complexity theory.

Inhalt

Relation among resources like time, space, determinism, nondeterminism, complexity classes, reduction and completeness, circuits and nonuniform complexity classes, logarithmic space and parallel complexity classes, Immerman-Szelepcsényi theorem, polynomial time hierarchy, relativization, parity and the polynomial methods, Valiant-Vazirani theorem, counting problems and classes, Toda's theorem, probabilistic computations, isolation lemma and parallel algorithms for matching, circuit identity testing, graph isomorphism and interactive proofs.

Literaturhinweise

Arora, Barak: Computational Complexity – A Modern Approach, Cambridge University Press
Oded Goldreich: Computational Complexity – A Conceptual Approach, Cambridge University Press
Dexter Kozen: Theory of Computation, Springer
Schöning, Pruim: Gems of Theoretical Computer Science, Springer

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Frank-Olaf Schreyer

Dozent/inn/en Prof. Dr. Frank-Olaf Schreyer

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Solving the exercises, passing the midterm and the final exam.

Lehrveranstaltungen / SWS

4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand

90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

Solving problems occurring in computer algebra praxis
The theory behind algorithms

Inhalt

Arithmetic and algebraic systems of equations in geometry, engineering and natural sciences

- integer and modular arithmetics, prime number tests
- polynomial arithmetics and factorization
- fast Fourier-transformation, modular algorithms
- resultants, Gröbnerbasen
- homotopy methods for numerical solving
- real solutions, Sturm chains and other rules for algebraic signs
- integer and modular arithmetics, prime number tests
- polynomial arithmetics and factorization
- fast Fourier-transformation, modular algorithms
- resultants, Gröbnerbasen
- homotopy methods for numerical solving
- real solutions, Sturm chains and other rules for algebraic signs

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Philipp Slusallek

Dozent/inn/en Prof. Dr. Philipp Slusallek

Zulassungsvoraussetzungen Solid knowledge of linear algebra is recommended.

Leistungskontrollen / Prüfungen

- Successful completion of weekly exercises (30% of final grade)
- Successful participation in rendering competition (10%)
- Mid-term written exam (20%, final exam prerequisite)
- Final written exam (40%)
- In each of the above a minimum of 50% is required to pass

A re-exam typically takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote The grade is derived from the above assessments. Possible changes will be announced at the beginning of each semester.

Sprache English

Lernziele / Kompetenzen

This course provides the theoretical and practical foundation for computer graphics. It gives a wide overview of topics, techniques, and approaches used in various aspects of computer graphics but has some focus on image synthesis or rendering. The first part of the course uses ray tracing as a driving applications to discuss core topics of computer graphics, from vector algebra all the way to sampling theory, the human visual system, sampling theory, and spline curves and surfaces. A second part then uses rasterization approach as a driving example, introducing the camera transformation, clipping, the OpenGL API and shading language, plus advanced techniques.

As part of the practical exercises the students incrementally build their own ray tracing system. Once the basics have been covered, the students participate in a rendering competition. Here they can implement their favorite advanced algorithm and are asked to generate a high-quality rendered image that shows their techniques in action.

Inhalt

- Introduction
- Overview of Ray Tracing and Intersection Methods
- Spatial Index Structures
- Vector Algebra, Homogeneous Coordinates, and Transformations
- Light Transport Theory, Rendering Equation
- BRDF, Materials Models, and Shading
- Texturing Methods
- Spectral Analysis, Sampling Theory
- Filtering and Anti-Aliasing Methods

- Recursive Ray Tracing & Distribution Ray-Tracing
- Human Visual System & Color Models
- Spline Curves and Surfaces
- Camera Transformations & Clipping
- Rasterization Pipeline
- OpenGL API & GLSL Shading
- Volume Rendering (opt.)

Literaturhinweise

Will be announced in the lecture.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Dr. Nico Döttling

Dozent/inn/en Prof. Dr. Cas Cremers
 Dr. Nico Döttling
 Dr. Antoine Joux
 Dr. Lucjan Hanzlik
 Dr. Julian Loss

Zulassungsvoraussetzungen For graduate students: Basic knowledge in theoretical computer science required, background knowledge in number theory and complexity theory helpful

Leistungskontrollen / Prüfungen

- Oral / written exam (depending on the number of students)
- A re-exam is normally provided (as written or oral examination).

Lehrveranstaltungen / SWS 4 h lectures
 + 2 h tutorial
 = 6 h (weekly)

Arbeitsaufwand 90 h of classes
 + 180 h private study
 = 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

The students will acquire a comprehensive knowledge of the basic concepts of cryptography and formal definitions. They will be able to prove the security of basic techniques.

Inhalt

- Symmetric and asymmetric encryption
- Digital signatures and message authentication codes
- Information theoretic and complexity theoretic definitions of security, cryptographic reduction proofs
- Cryptographic models, e.g. random oracle model
- Cryptographic primitives, e.g. trapdoor-one-way functions, pseudo random generators, etc.
- Cryptography in practice (standards, products)
- Selected topics from current research

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Martina Maggio

Dozent/inn/en Prof. Dr. Martina Maggio

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

- Written exam at the end of the course.
- A re-exam takes place before the start of the following semester.

Lehrveranstaltungen / SWS

4 h lectures
+ 2 h tutorials
= 6 h (weekly)

Arbeitsaufwand

75 h lectures
+ 15 h mandatory assignments
+ 180 h individual study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams and assignments. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

By completing the Cyber-Physical Systems course, students will acquire the ability to model, analyze, control, and implement embedded systems that interact with the physical world, equipping them to design reliable and efficient systems for a variety of applications in modern technology.

Inhalt

Cyber-Physical Systems are embedded systems that integrate computation with physical processes. These systems are ubiquitous in our daily lives, powering technologies such as smart watches, household appliances, mobile phones, and automotive control systems. In fact, the majority of modern computing devices are embedded systems, with an estimated 98% of new CPUs being embedded in larger systems.

This course provides a comprehensive foundation for understanding, designing, and programming cyber-physical systems, emphasizing their theoretical and practical aspects. It is structured into three interconnected parts:

1. *Models*: Students will learn how to represent the physical systems that embedded systems interact with, exploring dynamical systems in both continuous and discrete time. Additionally, the course will briefly introduce more advanced models, which combine discrete state systems with dynamical systems.
2. *Control*: This module focuses on principles for modifying the behavior of physical systems through computation. Students will study and apply control techniques such as state feedback and PID control, learning how these methods influence the interaction between embedded systems and their environments.
3. *Implementation*: The final course part addresses practical challenges in embedded systems programming. Topics include scheduling, communication, and fault tolerance. This ensures that students are equipped to implement robust and efficient embedded systems in real-world scenarios.

By the end of this course, students will possess the skills needed to design and implement cyber-physical systems that meet specific functional and performance requirements, preparing them for roles in cutting-edge industries where embedded systems play a critical role, such as the automotive industry and for research in the cyber-physical systems domain.

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Weitere Informationen

This module was formerly also known as *Embedded Systems*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr.-Ing. Holger Hermanns

Dozent/inn/en Prof. Dr.-Ing. Holger Hermanns
Prof. Dr. Anja Feldmann

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Qualification for final exam through mini quizzes during classes
- Possibility to get bonus points through excellent homework
- Final exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

After taking the course students have

- a thorough knowledge regarding the basic principles of communication networks,
- the fundamentals of protocols and concepts of protocol,
- Insights into fundamental motivations of different pragmatics of current network solutions,
- Introduction to practical aspects of data networks focusing on internet protocol hierarchies

Inhalt

Introduction and overview

Cross section:

- Stochastic Processes, Markov models,
- Fundamentals of data network performance assessment
- Principles of reliable data transfer
- Protocols and their elementary parts
- Graphs and Graphalgorithms (maximal flow, spanning tree)
- Application layer:
- Services and protocols
- FTP, Telnet
- Electronic Mail (Basics and Principles, SMTP, POP3, ..)
- World Wide Web (History, HTTP, HTML)

- Transport Layer:
 - Services and protocols
 - Addressing
 - Connections and ports
 - Flow control
 - QoS
- Transport Protocols (UDP, TCP, SCTP, Ports)
- Network layer:
 - Services and protocols
 - Routing algorithms
 - Congestion Control
 - Addressing
- Internet protocol (IP)
- Data link layer:
 - Services and protocols
 - Medium access protocols: Aloha, CSMA (-CD/CA), Token passing
 - Error correcting codes
 - Flow control
- Applications: LAN, Ethernet, Token Architectures, WLAN, ATM
- Physical layer
- Peer-to-Peer and Ad-hoc Networking Principles

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Jens Dittrich

Dozent/inn/en Prof. Dr. Jens Dittrich

Zulassungsvoraussetzungen especially Saarland University CS department's undergraduate lecture *Big Data Engineering* (former *Informationssysteme*), *Programmierung 1* and *2*, *Algorithmen und Datenstrukturen* as well as *Nebenläufige Programmierung*

For graduate students:

- motivation for databases and database management systems;
- the relational data model;
- relational query languages, particularly relational algebra and SQL;
- **solid** programming skills in Java and/or C++
- undergrad courses in algorithms and data structures, concurrent programming

Leistungskontrollen / Prüfungen

- Passing a two-hour written exam at the end of the semester
- Successful demonstration of programming project (teams of up to three students are allowed); the project may be integrated to be part of the weekly assignments

Grades are based on written exam; 50% in weekly assignments (in paper and additionally paper or electronic quizzes) must be passed to participate in the final and repetition exams.

A repetition exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

This class may be run as a flipped classroom, i.e. 2 hours of lectures may be replaced by self-study of videos/papers; the other 2 hours may be used to run a group exercise supervised by the professor called "the LAB")

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined based on project, midterm and best of endterm and reexam.

Sprache English

Lernziele / Kompetenzen

Database systems are the backbone of most modern information systems and a core technology without which today's economy – as well as many other aspects of our lives – would be impossible in their present forms. The course teaches the architectural and algorithmic foundations of modern database management systems (DBMS), focussing on database systems internals rather than applications. Emphasis is made on robust and time-tested techniques that have led databases to be considered a mature technology and one of the greatest success stories in computer science. At the same time, opportunities for exciting research in this field will be pointed out.

In the exercise part of the course, important components of a DBMS will be treated and where possible implemented and their performance evaluated. The goal this is to work with the techniques introduced in the lecture and to understand them and their practical implications to a depth that would not be attainable by purely theoretical study.

Inhalt

The course "Database Systems" will introduce students to the internal workings of a DBMS, in particular:

- storage media (disk, flash, main memory, caches, and any other future storage medium)
- data managing architectures (DBMS, streams, file systems, clouds, appliances)
- storage management (DB-file systems, raw devices, write-strategies, differential files, buffer management)
- data layouts (horizontal and vertical partitioning, columns, hybrid mappings, compression, defragmentation)
- indexing (one- and multidimensional, tree-structured, hash-, partition-based, bulk-loading and external sorting, differential indexing, read- and write-optimized indexing, data warehouse indexing, main-memory indexes, sparse and dense, direct and indirect, clustered and unclustered, main memory versus disk and/or flash-based)
- processing models (operator model, pipeline models, push and pull, block-based iteration, vectorization, query compilation)
- processing implementations (join algorithms for relational data, grouping and early aggregation, filtering)
- query processing (scanning, plan computation, SIMD)
- query optimization (query rewrite, cost models, cost-based optimization, join order, join graph, plan enumeration)
- data recovery (single versus multiple instance, logging, ARIES)
- parallelization of data and queries (horizontal and vertical partitioning, shared-nothing, replication, distributed query processing, NoSQL, MapReduce, Hadoop and/or similar and/or future systems)
- read-optimized system concepts (search engines, data warehouses, OLAP)
- write-optimized system concepts (OLTP, streaming data)
- management of geographical data (GIS, google maps and similar tools)
- main-memory techniques

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr.-Ing. Thorsten Herfet

Dozent/inn/en Prof. Dr.-Ing. Thorsten Herfet

Zulassungsvoraussetzungen The lecture requires a solid foundation of mathematics (differential and integral calculus) and probability theory. The course will, however, refresh those areas indispensably necessary for telecommunications and potential intensification courses and by this open this potential field of intensification to everyone of you.

Leistungskontrollen / Prüfungen Regular attendance of classes and tutorials
 Passing the final exam in the 2nd week after the end of courses.
 Eligibility: Weekly exercises / task sheets, grouped into two blocks corresponding to first and second half of the lecture. Students must provide min. 50% grade in each of the two blocks to be eligible for the exam.

Lehrveranstaltungen / SWS 4 h lectures
 + 2 h tutorial
 = 6 h (weekly)

Arbeitsaufwand 90 h of classes
 + 180 h private study
 = 270 h (= 9 ECTS)

Modulnote Final exam mark

Sprache English

Lernziele / Kompetenzen

Digital Signal Transmission and Signal Processing refreshes the foundation laid in "Signals and Systems" [Modulkennung]. Including, however, the respective basics so that the various facets of the introductory study period (Bachelor in Computer Science, Vordiplom Computer- und Kommunikationstechnik, Elektrotechnik or Mechatronik) and the potential main study period (Master in Computer Science, Diplom-Ingenieur Computer- und Kommunikationstechnik or Mechatronik) will be paid respect to.

Inhalt

As the basic principle, the course will give an introduction into the various building blocks that modern telecommunication systems do incorporate. Sources, sinks, source and channel coding, modulation and multiplexing are the major keywords, but we will also deal with dedicated pieces like A/D- and D/A-converters and quantizers in a little bit more depth.

The course will refresh the basic transformations (Fourier, Laplace) that give access to system analysis in the frequency domain, it will introduce derived transformations (z, Hilbert) for the analysis of discrete systems and modulation schemes and it will briefly introduce algebra on finite fields to systematically deal with error correction schemes that play an important role in modern communication systems.

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Weitere Informationen

This module was formerly also known as *Telecommunications I*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Peter Druschel, Ph.D.

Dozent/inn/en Prof. Peter Druschel, Ph.D.
Allen Clement, Ph.D

Zulassungsvoraussetzungen *Operating Systems or Concurrent Programming*

Leistungskontrollen / Prüfungen

- Regular attendance at classes and tutorials.
- Successful completion of a course project in teams of 2 students. (Project assignments due approximately every 2 weeks.)
- Passing grade on 2 out of 3 written exams: midterm, final exam, and a re-exam that takes place during the last two weeks before the start of lectures in the following semester.
- Final course grade: 50% project, 50% best 2 out of 3 exams.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

Introduction to the principles, design, and implementation of distributed systems.

Inhalt

- Communication: Remote procedure call, distributed objects, event notification, Inhalt dissemination, group communication, epidemic protocols.
- Distributed storage systems: Caching, logging, recovery, leases.
- Naming. Scalable name resolution.
- Synchronization: Clock synchronization, logical clocks, vector clocks, distributed snapshots.
- Fault tolerance: Replication protocols, consistency models, consistency versus availability trade-offs, state machine replication, consensus, Paxos, PBFT.
- Peer-to-peer systems: consistent hashing, self-organization, incentives, distributed hash tables, Inhalt distribution networks.
- Data centers. Architecture and infrastructure, distributed programming, energy efficiency.

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Hans-Peter Seidel

Dozent/inn/en Prof. Dr. Hans-Peter Seidel
Dr. Rhaleb Zayer

Zulassungsvoraussetzungen calculus and basic programming skills

Leistungskontrollen / Prüfungen

- Regular attendance and participation.
- Weekly Assignments (10% bonus towards the course grade; bonus points can only improve the grade; they do not affect passing)
- Passing the written exams (mid-term and final exam).
- The mid-term and the final exam count for 50% each, but 10% bonus from assignments will be added.
- A re-exam takes place at the end of the semester break or early in the next semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Practical assignments in groups of 3 students (practice)
Tutorials consists of a mix of theoretical + practical assignments.

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be based on the performance in exams, exercises and practical tasks. The detailed terms will be announced by the module coordinator.

Sprache English

Lernziele / Kompetenzen

Gaining knowledge of the theoretical aspect of geometric modelling problems, and the practical solutions used for modelling and manipulating curves and surfaces on a computer. From a broader perspective: Learning how to represent and interact with geometric models in a discretized, digital form (geometric representations by functions and samples; design of linear function spaces; finding “good” functions with respect to a geometric modelling task in such spaces).

Inhalt

- Differential geometry Fundamentals
- Interpolation and Approximation
- Polynomial Curves
- Bezier and Rational Bezier Curves
- B-splines, NURBS
- Spline Surfaces
- Subdivision and Multiresolution Modelling
- Mesh processing
- Approximation of differential operators
- Shape Analysis and Geometry Processing

Literaturhinweise

Will be announced before the term begins on the lecture website.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Jürgen Steimle

Dozent/inn/en Prof. Dr. Jürgen Steimle

Zulassungsvoraussetzungen undergraduate students: *Programmierung 1* and *2*
graduate students: none

Leistungskontrollen / Prüfungen Regular attendance of classes and tutorials
Successful completion of exercises and course project
Final exam
A re-exam takes place (as written or oral examination).

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

This course teaches the theoretical and practical foundations for human computer interaction. It covers a wide overview of topics, techniques and approaches used for the design and evaluation of modern user interfaces.

The course covers the principles that underlie successful user interfaces, provides an overview of input and output devices and user interface types, and familiarizes students with the methods for designing and evaluating user interfaces. Students learn to critically assess user interfaces, to design user interfaces themselves, and to evaluate them in empirical studies.

Inhalt

- Fundamentals of human-computer interaction
- User interface paradigms, input and output devices
- Desktop & graphical user interfaces
- Mobile user interfaces
- Natural user interfaces
- User-centered interaction design
- Design principles and guidelines
- Prototyping

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Joachim Weickert

Dozent/inn/en Prof. Dr. Joachim Weickert

Zulassungsvoraussetzungen Undergraduate mathematics (e.g. Mathematik für Informatiker I-III) and elementary programming knowledge in C

Leistungskontrollen / Prüfungen

- For the homework assignments one can obtain up to 24 points per week. Actively participating in the classroom assignments gives 12 more points per week, regardless of the correctness of the solutions. To qualify for both exams one needs 2/3 of all possible points.
- Passing the final exam or the re-exam.
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from the performance in the exam or the re-exam. The better grade counts.

Sprache English

Lernziele / Kompetenzen

Broad introduction to mathematical methods in image processing and computer vision. The lecture qualifies students for a bachelor thesis in this field. Together with the completion of advanced or specialised lectures (9 credits at least) it is the basis for a master thesis in this field.

Inhalt

Inhalt

1. Basics
 - 1.1 Image Types and Discretisation
 - 1.2 Degradations in Digital Images
2. Colour Perception and Colour Spaces
3. Image Transformations
 - 3.1 Continuous Fourier Transform
 - 3.2 Discrete Fourier Transform
 - 3.3 Image Pyramids
 - 3.4 Wavelet Transform
4. Image Compression
5. Image Interpolation
6. Image Enhancement
 - 6.1 Point Operations

- 6.2 Linear Filtering and Feature Detection
- 6.3 Morphology and Median Filters
- 6.3 Wavelet Shrinkage, Bilateral Filters, NL Means
- 6.5 Diffusion Filtering
- 6.6 Variational Methods
- 6.7 Deconvolution Methods
- 7. Texture Analysis
- 8. Segmentation
 - 8.1 Classical Methods
 - 8.2 Variational Methods
- 9. Image Sequence Analysis
 - 9.1 Local Methods
 - 9.2 Variational Methods
- 10. 3-D Reconstruction
 - 10.1 Camera Geometry
 - 10.2 Stereo
 - 10.3 Shape-from-Shading
- 11. Object Recognition
 - 11.1 Hough Transform
 - 11.2 Invariants
 - 11.3 Eigenspace Methods

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Gerhard Weikum

Dozent/inn/en Prof. Dr. Gerhard Weikum

Zulassungsvoraussetzungen Good knowledge of undergraduate mathematics (linear algebra, probability theory) and basic algorithms.

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutor groups
- Presentation of solutions in tutor groups
- Passing 2 of 3 written tests (after each third of the semester)
- Passing the final exam (at the end of the semester)

Lehrveranstaltungen / SWS

4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand

90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined by the performance in written tests, tutor groups, and the final exam. Details will be announced on the course web site.

Sprache English

Lernziele / Kompetenzen

The lecture teaches models and algorithms that form the basis for search engines and for data mining and data analysis tools.

Inhalt

Information Retrieval (IR) and Data Mining (DM) are methodologies for organizing, searching and analyzing digital contents from the web, social media and enterprises as well as multivariate datasets in these contexts. IR models and algorithms include text indexing, query processing, search result ranking, and information extraction for semantic search. DM models and algorithms include pattern mining, rule mining, classification and recommendation. Both fields build on mathematical foundations from the areas of linear algebra, graph theory, and probability and statistics.

Literaturhinweise

Will be announced on the course web site.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Gert Smolka

Dozent/inn/en Prof. Dr. Gert Smolka

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials.
- Passing the midterm and the final exam.

Lehrveranstaltungen / SWS

4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand

90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

- structure of logic languages based on type theory
- distinction notation / syntax / semantics
- structure and formal representation of mathematical statements
- structure and formal representation of proofs (equational and natural deduction)
- solving Boolean equations
- proving formulas with quantifiers
- implementing syntax and deduction

Inhalt

Type Theory:

- functional representation of mathematical statements
- simply typed lambda calculus, De Bruijn representation and substitution, normalization, elimination of lambdas
- Interpretations and semantic consequence
- Equational deduction, soundness and completeness
- Propositional Logic
- Boolean Axioms, completeness for 2-valued interpretation
- resolution of Boolean equations, canonical forms based on decision trees and resolution

Predicate Logic (higher-order):

- quantifier axioms
- natural deduction
- prenex and Skolem forms

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Isabel Valera

Dozent/inn/en Prof. Dr. Isabel Valera

Zulassungsvoraussetzungen The lecture gives a broad introduction into machine learning methods. After the lecture the students should be able to solve and analyze learning problems.

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials.
- 50% of all points of the exercises have to be obtained in order to qualify for the exam.
- Passing 1 out of 2 exams (final, re-exam).

Lehrveranstaltungen / SWS

4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand

90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Determined from the results of the exams, exercises and potential projects. The exact grading modalities are announced at the beginning of the course.

Sprache English

Lernziele / Kompetenzen

The lecture gives a broad introduction into machine learning methods. After the lecture the students should be able to solve and analyze learning problems.

Inhalt

- Bayesian decision theory
- Linear classification and regression
- Kernel methods
- Bayesian learning
- Semi-supervised learning
- Unsupervised learning
- Model selection and evaluation of learning methods
- Statistical learning theory
- Other current research topics

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Peter Druschel, Ph.D.

Dozent/inn/en Prof. Peter Druschel, Ph.D.
Björn Brandenburg, Ph.D

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen Regular attendance at classes and tutorials
Successful completion of a course project in teams of 2 students
Passing 2 written exams (midterm and final exam)
A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

Introduction to the principles, design, and implementation of operating systems

Inhalt

Process management:

- Threads and processes, synchronization
- Multiprogramming, CPU Scheduling
- Deadlock

Memory management:

- Dynamic storage allocation
- Sharing main memory
- Virtual memory

I/O management:

- File storage management
- Naming
- Concurrency, Robustness, Performance

Virtual machines

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Kurt Mehlhorn

Dozent/inn/en Prof. Dr. Kurt Mehlhorn
Dr. Andreas Karrenbauer

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Solving accompanying exercises, successful participation in midterm and final exam
- Grades: Yes
- The grade is calculated from the above parameters according to the following scheme: 20%, 30%, 50%
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

The students learn to model and solve optimization problems from theory as from the real world

Inhalt

Linear Programming: Theory of polyhedra, simplex algorithm, duality, ellipsoid method * Integer linear programming: Branch-and-Bound, cutting planes, TDI-Systems * Network flow: Minimum cost network flow, minimum mean cycle cancellation algorithm, network simplex method * Matchings in graphs: Polynomial matching algorithms in general graphs, integrality of the matching polytope, cutting planes * Approximation algorithms: LP-Rounding, greedy methods, knapsack, bin packing, steiner trees and forests, survivable network design

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Michael Backes

Dozent/inn/en Prof. Dr. Michael Backes
Prof. Dr. Cas Cremers

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Passing the final exam
- A re-exam is normally provided (as written or oral examination).

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined by the performance in exams, tutor groups, and practical tasks.
Details will be announced by the lecturer at the beginning of the course.

Sprache English

Lernziele / Kompetenzen

Description, assessment, development and application of security mechanisms, techniques and tools.

Inhalt

- Basic Cryptography,
- Specification and verification of security protocols,
- Security policies: access control, information flow analysis,
- Network security,
- Media security,
- Security engineering

Literaturhinweise

Will be announced on the course website

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Gert Smolka

Dozent/inn/en Prof. Dr. Gert Smolka

Zulassungsvoraussetzungen For graduate students: core lecture Introduction to Computational Logic

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials.
- Passing the midterm and the final exam

Lehrveranstaltungen / SWS

4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand

90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

Understanding of

- Logical structure of programming languages
- Formal models of programming languages
- Type and module systems for programming languages

Inhalt

Theory of programming languages, in particular:

- Formal models of functional and object-oriented languages
- Lambda Calculi (untyped, simply typed, System F, F-omega, Lambda Cube, subtyping, recursive types, Curry-Howard Correspondence)
- Algorithms for type checking and type reconstruction

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Sven Apel

Dozent/inn/en Prof. Dr. Sven Apel

Zulassungsvoraussetzungen

- Knowledge of programming concepts (as taught in the lectures *Programmierung 1* and *Programmierung 2*)
- Basic knowledge of software processes, design, and testing (as taught and applied in the lecture *Softwarepraktikum*)

Leistungskontrollen / Prüfungen Beside the lecture and weekly practical exercises, there will be a number of assignments in the form of mini-projects for each student to work on (every two to three weeks). The assignments will be assessed based on the principles covered in the lecture. Passing all assignments is a prerequisite for taking the final written exam. The final grade is determined only by the written exam. Further examination details will be announced by the lecturer at the beginning of the course. In short:

- Passing all assignments (prerequisite for the written exam)
- Passing the written exam

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h exercises
= 6 h (weekly)

Arbeitsaufwand 90 h of classes and exercises
+ 180 h private study and assignments
= 270 h (= 9 ECTS)

Modulnote The grade is determined by the written exam. Passing all assignments is a prerequisite for taking the written exam. The assignments do not contribute to the final grade. Further examination details will be announced by the lecturer at the beginning of the course.

Sprache English

Lernziele / Kompetenzen

- The students know and apply modern software development techniques.
- They are aware of key factors contributing to the complexity of real-world software systems, in particular, software variability, configurability, feature interaction, crosscutting concerns, and how to address them.
- They know how to apply established design and implementation techniques to master software complexity.
- They are aware of advanced design and implementation techniques, including collaboration-based design, mixins/traits, aspects, pointcuts, advice.
- They are aware of advanced quality assurance techniques that take the complexity of real-world software systems into account: variability-aware analysis, sampling, feature-interaction detection, predictive performance modeling, etc.
- They appreciate the role of non-functional properties and know how to predict and optimize software systems regarding these properties.
- They are able to use formal methods to reason about key techniques and properties covered in the lecture.

Inhalt

- Domain analysis, feature modeling
- Automated reasoning about software configuration using SAT solvers
- Runtime parameters, design patterns, frameworks
- Version control, build systems, preprocessors
- Collaboration-based design
- Aspects, pointcuts, advice
- Expression problem, preplanning problem, code scattering & tangling, tyranny of the dominant decomposition, inheritance vs. delegation vs. mixin composition
- Feature interaction problem (structural, control- & data-flow, behavioral, non-functional feature interactions)
- Variability-aware analysis and variational program representation (with applications to type checking and static program analysis)
- Sampling (random, coverage)
- Machine learning for software performance prediction and optimization

Literaturhinweise

- Feature-Oriented Software Product Lines: Concepts and Implementation. S. Apel, et al., Springer, 2013.
- Generative Programming: Methods, Tools, and Applications: Methods, Techniques and Applications. K. Czarnecki, et al., Addison-Wesley, 2000.
- Mastering Software Variability with FeatureIDE. J. Meinicke, et al., Springer, 2017.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr.-Ing. Holger Hermanns

Dozent/inn/en Prof. Dr.-Ing. Holger Hermanns
Prof. Bernd Finkbeiner, Ph.D

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Passing the final exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS 4 h lectures
+ 2 h tutorial
= 6 h (weekly)

Arbeitsaufwand 90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)

Modulnote Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

The students become familiar with the standard methods in computer-aided verification. They understand the theoretical foundations and are able to assess the advantages and disadvantages of different methods for a specific verification project. The students gain first experience with manual correctness proofs and with the use of verification tools.

Inhalt

- models of computation and specification languages: temporal logics, automata over infinite objects, process algebra
- deductive verification: proof systems (e.g., Floyd, Hoare, Manna/Pnueli), relative completeness, compositionality
- model checking: complexity of model checking algorithms, symbolic model checking, abstraction case studies

Literaturhinweise

Will be announced before the start of the course on the course page on the Internet.