



UNIVERSITÄT
DES
SAARLANDES

FAKULTÄT FÜR MATHEMATIK UND INFORMATIK

MODULHANDBUCH

Quereinstiegsmaster Lehramt Informatik

21. Oktober 2024

Liste der Modulbereiche und Module

1	Pflichtbereich	3
1.1	Ausgewählte Themen für das Lehramt Informatik (Q-Master)	4
1.2	Einführung in die Didaktik der Informatik	5
1.3	Elementare fachdidaktische schulpraktische Studien	7
1.4	Fachdidaktikseminar I	9
1.5	Fachdidaktikseminar II	10
1.6	Masterarbeit (Q-Master Lehramt Informatik)	11
1.7	Masterseminar (Q-Master Lehramt Informatik)	12
1.8	Vertiefende fachdidaktische schulpraktische Studien	13
2	Wahlpflichtbereich 1	14
2.1	Algorithms and Data Structures	15
2.2	Artificial Intelligence	16
2.3	Cryptography	18
2.4	Data Networks	19
2.5	Database Systems	21
2.6	Elements of Data Science and Artificial Intelligence	23
2.7	Foundations of Cybersecurity 1	24
2.8	Human Computer Interaction	25
2.9	Image Processing and Computer Vision	26
2.10	Introduction to Computational Logic	28
2.11	Machine Learning	29
2.12	Security	30
2.13	Software Engineering	31
3	Wahlpflichtbereich 2	33
3.1	Big Data Engineering	34
3.2	Elements of Machine Learning	37
3.3	Ethics for Nerds	39
3.4	Nebenläufige Programmierung	41

3.5	Systemarchitektur für Informatik-Lehramt	44
-----	--	----

Modulbereich 1

Pflichtbereich

Ausgewählte Themen für das Lehramt Informatik (Q-Master)

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
		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)

Arbeitsaufwand 45 Stunden Präsenzzeit + 75 Stunden Selbststudium = 120 Stunden (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 und künstlicher Intelligenz
- Grundlagen der Digitalen Ethik

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
	1-3	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
	2	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
	1-3	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
	2-4	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.

Masterarbeit (Q-Master Lehramt Informatik)

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	4	Se.	1 Semester	-	22

Modulverantwortliche/r Prof. Dr. Verena Wolf

Dozent/inn/en Professoren der Fachrichtung

Zulassungsvoraussetzungen Erfolgreicher Abschluss des Masterseminars

Leistungskontrollen / Prüfungen Eigenständige Bearbeitung des Vorhabens, Wissenschaftliches Paper und Kolloquiumsvortrag

Lehrveranstaltungen / SWS •

Arbeitsaufwand 40 Stunden Kontaktzeit mit Betreuer + 620 Stunden Selbststudium = 660 Stunden (22 ECTS)

Modulnote Ergibt sich aus der Bewertung der Masterarbeit

Sprache deutsch

Lernziele / Kompetenzen

Die Masterarbeit zeigt die Fähigkeiten der oder des Studierenden zur eigenständigen Durchführung wissenschaftlicher Forschung in einem angemessenen und im Masterseminar vorbereiteten Themengebiet.

Inhalt

Die Masterarbeit zeigt die Fähigkeiten der oder des Studierenden zur eigenständigen Durchführung wissenschaftlicher Forschung in einem angemessenen und im Masterseminar vorbereiteten Themengebiet.

Literaturhinweise

Masterseminar (Q-Master Lehrmat Informatik)

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	3	Se.	1 Semester	2	6

Modulverantwortliche/r Prof. Dr. Verena Wolf

Dozent/inn/en Prof. Dr. Verena Wolf

Zulassungsvoraussetzungen 12 CP im Bereich Informatik

Leistungskontrollen / Prüfungen

- Vorbereitung relevanter Fachliteratur
- Schriftliche Ausarbeitung des Themas der Masterarbeit
- Präsentation mit anschließender Diskussion zum geplanten Vorhaben
- Aktive Teilnahme an Diskussionen

Lehrveranstaltungen / SWS Seminar (2 SWS)

Arbeitsaufwand 20 h Seminar + 20 h Kontaktzeit mit Betreuer + 140 h Selbststudium = 180 h (6 ECTS)

Modulnote ergibt sich aus der Leistung in Vortrag und Ausarbeitung

Sprache englisch

Lernziele / Kompetenzen

Im Master Seminar wird die Grundlage für die eigenständige Auseinandersetzung mit dem Forschungsvorhaben der Masterarbeit gelegt, das von den Teilnehmenden zu einem wesentlichen Teil mitbestimmt werden kann.

Am Ende des Seminars wurden einerseits Fähigkeiten und Fertigkeiten zur erfolgreichen Bearbeitung der Masterarbeit erlernt sowie wissenschaftliche Methoden zur Bearbeitung des Projekts festgelegt.

Das Masterseminar bereitet in diesem Sinne auf die Bearbeitung der Masterarbeit sowie wissenschaftlichen Diskurs vor. Dazu ist eine aktive Teilnahme an Diskussionen während des Seminars notwendig.

Inhalt

Anwendung wissenschaftlicher Methoden der Informatik nach aktuellem Forschungsstand.

Literaturhinweise

Wird in Absprache mit Betreuer und Dozent individuell festgelegt.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
3		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.

Modulbereich 2

Wahlpflichtbereich 1

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	1-3	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
	1-3	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
	1-3	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
	1-3	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
	1-3	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
	1-3	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
	1-3	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
	1-3	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
	1-3	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
	1-3	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
	1-3	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
	1-3	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
	1-3	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.

Modulbereich 3

Wahlpflichtbereich 2

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	1-3	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
	1-3	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
	1-3	occasional / summer semester	1 semester	4	6

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

Dozent/inn/en Prof. Dr.-Ing. Holger Hermanns
Kevin Baum
Sarah Sterz

Zulassungsvoraussetzungen We expect basic knowledge of propositional and first-order logic, an open mind, and interest to look at computer science in ways you probably are not used to.

Leistungskontrollen / Prüfungen The details of exam admission and grading are announced at the beginning of each iteration. Typically, participant are graded based on

- an exam or a re-exam (the better mark counts),
- a short essay where the participant has to argue for or against a moral claim in a topic from computer science.

To get the exam admission, participants usually have to get 50% of the points on weekly exercise sheets.

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

(may be adjusted before the start of each iteration of the course)

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

Modulnote Will be determined based on exam performance, essay performance, and possibly exercise outcomes. The exact modalities will be announced at the beginning of the module.

Sprache English

Lernziele / Kompetenzen

Many computer scientists will be confronted with morally difficult situations at some point in their career – be it in research, in business, or in industry. This module equips participants with the crucial assets enabling them to recognize such situations and to devise ways to arrive at a justified moral judgment regarding the question what one is permitted to do and what one should better not do. For that, participants will be made familiar with moral theories from philosophy, as well as different Codes of Ethics for computer scientists. Since one can quickly get lost when talking about ethics and morals, it is especially important to talk and argue clearly and precisely. In order to do prepare for that, the module offers substantial training regarding formal and informal argumentation skills enabling participants to argue beyond the level of everyday discussions at bars and parties. In the end, succesful participants are able to assess a morally controversial topic from computer science on their own and give a convincing argument for their respective assessments.

The module is intended to always be as clear, precise, and analytic as possible. What you won't find here is the meaningless bla-bla, needlessly poetic language, and vague and wordy profundity that some people tend to associate with philosophy.

Inhalt

This course covers:

- an introduction to the methods of philosophy, argumentation theory, and the basics of normative as well as applied ethics;
- relevant moral codices issued by professional associations like the ACM, the IEEE, and more;
- starting points to evaluate practices and technologies already in use or not that far away, including for instance: filter bubbles and echo chambers, ML-algorithms as predictive tools, GPS-tracking, CCTV and other tools from surveillance, fitness trackers, big data analysis, autonomous vehicles, lethal autonomous weapons systems and so on;
- an outlook on more futuristic topics like machine ethics, roboethics, and superintelligences;
- and more.

The content of the course is updated regularly to always be up-to-date and cover the currently most relevant topics, technologies, policies, and developments.

Literaturhinweise

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

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	1-3	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.
Nachklausuren finden innerhalb der letzten Wochen vor dem Vorlesungsbeginn des Folgesemesters statt.

Lehrveranstaltungen / SWS **Element T – Theorie (2 SWS):**
8 Vorlesungen: 6 Wochen
4 Übungen: 6 Wochen
Element A – Anwendung (2 SWS):
9 Vorlesungen: 6 Wochen
4 Übungen: 6 Wochen
Element P – Praxis (Eigenstudium):
Semesterbegleitend 8 schriftliche Reflektionen (Prüfungsvorleistungen),
anschließend Projektarbeit über ca. 2 Wochen
= 4 SWS

Arbeitsaufwand **Element T:**
24 h Präsenz, 36 h Selbststudium
Element A:
26 h Präsenz, 34 h Selbststudium
Element P:
60 h Selbststudium

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

Modulnote Wird aus Leistungen in Klausuren (im Anschluss an die Elemente T und A), sowie den Prüfungsvorleistungen (Element P) ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben. Alle Modulelemente sind innerhalb eines Prüfungszeitraumes erfolgreich zu absolvieren.

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 Go 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*.

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
	1-3	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