

Modularisierung der Lehramtsstudiengänge
Fach Informatik

Leitbild

Informatiklehrerinnen und – Lehrer sind Expertinnen und Experten für gezielte und nach wissenschaftlichen Erkenntnissen gestaltete Vermittlungs-, Lern- und Bildungsprozesse im Fach Informatik.

- Sie sehen fachliches Wissen und Verständnis als Beitrag zur Orientierung und Handlungsfähigkeit der Schülerinnen und Schüler in einer durch Informations- und Kommunikationstechnik geprägten Welt.
- Sie verfügen über anschlussfähiges Fachwissen, das es ihnen ermöglicht, Unterrichtskonzepte und –medien inhaltlich zu bewerten, aktuelle Forschung in der Informatik und ihren Handlungsfeldern zu verfolgen und neue Themen in den Unterricht einzubringen.
- Sie kennen die Ideengeschichte wichtiger Konzepte der Informatik. Wissenschaftliche und anwendungstechnische Aspekte sind integraler Bestandteil ihres Unterrichts.
- Sie können die gesellschaftliche Bedeutung der Informatik begründen und vertreten sie reflektiert im Unterricht und in der (Schul-)Öffentlichkeit.
- Sie orientieren ihr unterrichtliches Handeln an den Erkenntnissen der Fachdidaktik und der Bildungswissenschaften. Sie wissen, dass Schülerinnen und Schüler das Verständnis informatischer Konzepte selbst entwickeln müssen und fördern zielorientiertes selbsttätiges Lernen.
- Sie können bei Schülerinnen und Schülern Freude und Interesse für das Fach Informatik wecken und sie zu einem sachgerechten Umgang mit Informatiksystemen anleiten.

Kompetenzen

Aus diesem Leitbild werden folgende Kompetenzen abgeleitet, über die die Studierenden nach Abschluss ihres Studiums verfügen sollen.

1. Übergreifende Kompetenzen

- den Unterricht wissenschaftlich begründen und effektiv gestalten,
- grundlegende allgemeine wie fachspezifische Unterrichtsprinzipien beherrschen und anwenden,
- Methoden und Medien in ihrer systematischen und funktionalen Ordnung und Beziehung verstehen und adäquat anwenden bzw. einsetzen,
- Lernvoraussetzungen und Lernverhalten der Schülerinnen und Schüler einschätzen und ihnen Rechnung tragen,
- Lernprozesse erfolgreich gestalten,
- die eigene didaktische, methodische und pädagogische Arbeit selbstkritisch reflektieren.

2. Fachliche Kompetenzen

- über fundierte Kenntnisse insbesondere aus folgenden Informatikgebieten sowie über deren Zusammenhänge verfügen: formale Grundlagen (Algorithmusbegriff, formale Sprachen, mathematische Methoden zur Beurteilung von Informatiksystemen und deren Grenzen), Softwareentwicklung (Modellierung, Programmkonzepte, Algorithmen und Datenstrukturen, Softwareentwicklungsprozesse), Rechnerhardware (Aufbau und Funktionsweise der Hardware, Ausführung von Befehlen und Programmen durch den Prozessor),
- zentrale Fragestellungen der Informatik und damit verbundene Erkenntnisinteressen skizzieren sowie fachliche Fragen selbst entwickeln,
- Methoden der Informatik beschreiben und anwenden und sie hinsichtlich ihrer Möglichkeiten und Grenzen einschätzen,
- informatikbezogene Theorien und Prozesse der Begriffs- und Modellbildung erläutern und ihren Stellenwert reflektieren,
- Forschungsergebnisse der Informatik in ihrer fachlichen Bedeutung und Reichweite einschätzen,
- sich in neue Entwicklungen der Informatik in selbstständiger Weise einarbeiten,
- informatische Inhalte hinsichtlich ihrer gesellschaftlichen und historischen Bedeutung einordnen und Verbindungslinien zu anderen Wissenschaften aufzeigen,
- die Relevanz der fachlichen Fragestellungen, Methoden, theoretischen Ansätze und Forschungsergebnisse und Inhalte der Informatik in Bezug auf das spätere Berufsfeld Schule einschätzen.

3. Fachdidaktische Kompetenzen

- den Bildungsauftrag des Faches Informatik konkretisieren und den Informatikunterricht entsprechend ausrichten,
- den bildenden Gehalt informatischer Inhalte und Methoden reflektieren, informatische Inhalte in einen unterrichtlichen Zusammenhang bringen und durchdenken sowie fachübergreifende Perspektiven berücksichtigen,
- inhaltliche Schwerpunkte des Informatikunterrichts unter Berücksichtigung fachdidaktischer Forschungsergebnisse sinnvoll setzen,
- den Informatikunterricht in den curricularen Rahmen einordnen bzw. aus diesem ableiten,
- Inhalte des Informatikunterrichts bestimmen, didaktisch reduzieren und strukturieren,
- wissenschaftliche Fragestellungen und Sachverhalte der Informatik angemessen sach- und adressatenbezogen darstellen und präsentieren sowie hinsichtlich ihrer didaktischen Relevanz einordnen,
- Informatikunterricht unter Verwendung geeigneter Medien sowie Informations- und Kommunikationstechnologien analysieren, planen, erproben und reflektieren,
- Grundlagen und Prozesse fachlichen und fachübergreifenden Lernens in der Informatik unter Berücksichtigung fachspezifischer Lernschwierigkeiten und Fördermöglichkeiten analysieren und exemplarisch erläutern,
- fachrelevante Wege zur Lernerfolgskontrolle beherrschen,
- fachliche, fachübergreifende sowie fächerverbindende Sichtweisen in die Entwicklung von Schulprofilen und Schulprogrammen einbringen und die Bedeutung des Unterrichtsfaches Informatik im Kontext der Schulfächer sowie die Rolle als Informatiklehrerin oder Informatiklehrer reflektieren,
- die Bedeutung informatischer Bildung sowie Aufgabe und Aufbau des Informatikunterrichts in der (Schul-)Öffentlichkeit überzeugend und nachvollziehbar darlegen.

Modulübersicht und Modulhandbuch

Informatik – Lehramt an Gymnasien und Lehramt an Berufsbildenden Schulen

Studienabschnitt	Modul	Titel	Derzeitiger Modulverantwortlicher	ECTS	Studiengang Lehrämter
Grundstudium	CS 101	Ringvorlesungen Perspektiven der Informatik	Studiendekan der Fakultät Mathematik und Informatik bzw. Studienbeauftragter der Informatik	2	LAG, LAB
Grundstudium	IT - English	Sprachkurs IT - English	Dr. Peter Tischer	2	LAG, LAB
Grundstudium	CS 120 / P1	Programmierung 1	Prof. Dr. Gert Smolka	9	LAG, LAB
Grundstudium	CS 220 / P2	Programmierung 2	Prof. Dr. Hans-Peter Lenhof	9	LAG, LAB
Grundstudium	CS 230 / SysArch	Systemarchitektur	Prof. Dr. W.-J. Paul	9	LAG, LAB
Grundstudium	CS 110 / Mfi 1	Mathematik für Informatiker 1	Prof. Dr. Joachim Weickert	9	LAB
Grundstudium	CS 210 / Mfi 2	Mathematik für Informatiker 2	Prof. Dr. Joachim Weickert	9	LAB
Grundstudium	CS 340 / GrADS	Grundzüge von Algorithmen und Datenstrukturen	Prof. Dr. Raimund Seidel	6	LAG, LAB
Grundstudium	CS 330 / InfoSys	Informationssysteme	Prof. Dr. Gerhard Weikum	6	LAG, LAB
Grundstudium	CS 430	Nebenläufige Programmierung	Prof. Dr. Ing. Holger Hermanns	6	LAG, LAB
Grundstudium	CS CS 420 / TheoInf	Grundzüge der Theoretischen Informatik	Prof. Dr. Raimund Seidel	9	LAG, LAB
Grundstudium	CS CS 320 / SoDePra	Softwaredesign- praktikum	Prof. Dr. Andreas Zeller	9	LAG, LAB
Grundstudium	DDI	Didaktik der Informatik	Lehrbeauftragte/r für die Didaktik der Informatik	9	LAG, LAB
Grundstudium	EPS	Elementare fachdidaktische schulpraktische Studien	Lehrbeauftragte/r für die Didaktik der Informatik	7	LAG, LAB
Grundstudium	VPS	Vertiefte fachdidaktische schulpraktische Studien	Lehrbeauftragte/r für die Didaktik der Informatik	9	LAG, LAB
Hauptstudium	CS 500	Seminar	Studiendekan der Fakultät Mathematik und Informatik/ Studienbeauftragter der Informatik	7	LAG, LAB
Hauptstudium	CS 551 / OS	Operating Systems	Prof. Dr. Helge Scheidig	9	LAG, LAB
Hauptstudium	CS 552 / CG	Computer Graphics	Prof. Dr. Philipp Slusallek	9	LAG, LAB
Hauptstudium	CS 553 / DBS	Database Systems	Prof. Dr. Christoph Koch	9	LAG, LAB
Hauptstudium	CS 554 /DN	Data Networks	Prof. Dr. Holger Hermanns	9	LAG, LAB

Studien- abschnitt	Modul	Titel	Derzeitiger Modulverantwortlicher	ECTS	Studiengang Lehrämter
Hauptstudium	CS 555 / IRDM	Information Retrieval and Data Mining,	Prof. Dr. Gerhard Weikum	9	LAG, LAB
Hauptstudium	CS 556 / AI	Artificial Intelligence	Prof. Dr. Wolfgang Wahlster	9	LAG, LAB
Hauptstudium	CS 558 / CAR	Computer Architecture	Prof. Dr. W.-J. Paul	9	LAG, LAB
Hauptstudium	CS 559 / SEC	Security	PD Dr. Werner Stephan	9	LAG, LAB
Hauptstudium	CS 560 / SE	Software Engineering,	Prof. Dr. Andreas Zeller	9	LAG, LAB
Hauptstudium	CS 561 / CC	Compiler Construction	Prof. Dr. Reinhard Wilhelm	9	LAG, LAB
Hauptstudium	CS 571 / AR	Automated Reasoning	Prof. Dr. Bernd Finkbeiner, Prof. Dr. Jörg Siekmann	9	LAG, LAB
Hauptstudium	CS 572	Image Processing and Computer Vision	Prof. Dr. Joachim Weickert	9	LAG, LAB
Hauptstudium	CS 573	Computer Algebra	Prof. Dr. Frank-Olaf Schreyer	9	LAG, LAB
Hauptstudium	CS 574	Algorithms and Data Structures	Prof. Dr. Kurt Mehlhorn	9	LAG, LAB
Hauptstudium	CS 575	Introduction to Computational Logic	Prof. Dr. Gert Smolka	9	LAG, LAB
Hauptstudium	CS 576	Geometrische Modellierung	Prof. Dr. Hans-Peter Seidel	9	LAG, LAB
Hauptstudium	CS 577	Komplexitätstheorie	Prof. Dr. Markus Bläser	9	LAG, LAB
Hauptstudium	CS 578	Kryptographie	Prof. Dr. Michael Backes	9	LAG, LAB
Hauptstudium	CS 579	Optimierung	Studiendekan bzw. Studienbeauftragter der FR	9	LAG, LAB
Hauptstudium	CS 580	Semantics	Prof. Dr. Gert Smolka	9	LAG, LAB
Hauptstudium	CS 650/ECG	Effective Computational Geometry for Curves and Surfaces	Prof. Dr. Kurt Mehlhorn	9	LAG, LAB
Hauptstudium	CS 650/XXL	Algorithms For Large Data Sets	Prof. Dr. Kurt Mehlhorn	9	LAG, LAB
Hauptstudium	CS 650/I2A	Intelligent Information Agents for the Internet and Web	Prof. Dr. Wolfgang Wahlster	9	LAG, LAB
Hauptstudium	CS 650/MAS	Mathematical Assistant Systems	Prof. Dr. Jörg Siekmann, Dr. Christoph Benzmüller	9	LAG, LAB
Hauptstudium	CS 750/NLG	Natural Language Generation	Prof. Dr. Jörg Siekmann,	9	LAG, LAB
Hauptstudium	CS 750/ET	Educational Technologies	PD Dr. Erica Melis	9	LAG, LAB
Hauptstudium	CS 650 / CAR2	Computer Architecture 2	Prof. Dr. W. J. Paul	9	LAG, LAB
Hauptstudium	CS 650 / TC II	Telecommunications II	Prof. Dr.-Ing. Thorsten Herfet	9	LAG, LAB
Hauptstudium	CS 650 / DBT	Database Theory	Prof. Dr. Christoph Koch	6	LAG, LAB
Hauptstudium	CS 650 / AG&V	Automata, Games and Verification	Prof. Dr. Bernd Finkbeiner	6	LAG, LAB

Studien- abschnitt	Modul	Titel	Derzeitiger Modulverantwortlicher	ECTS	Studiengang Lehrämter
Hauptstudium	CS 650 / SEMHOL	Semantics and Mechanization of Classical Higher- Order Logic	Prof. Dr. Jörg Siekmann	9	LAG, LAB
Hauptstudium	CS 650 / AutoD	Automated Debugging	Prof. Dr. Andreas Zeller	6	LAG, LAB
Hauptstudium	CS 650 / ES	Embedded Systems	Prof. Dr. Andreas Zeller	9	LAG, LAB
Hauptstudium	CS 650 / DNII&VII	Data Networks II / Verification II	Prof. Dr.-Ing. Holger Hermanns	6	LAG, LAB
Hauptstudium	CS 750 / OSP	Operating Systems Practice	Prof. Dr. Helge Scheidig	4	LAG, LAB
Hauptstudium	CS 750 / DOS	Distributed Operating Software	Prof. Dr. Helge Scheidig	4	LAG, LAB
Hauptstudium	CS 650 / CGII-RIS	Computer Graphics II	Prof. Dr. Philipp Slusallek	9	LAG, LAB
Hauptstudium	CS 650 / CG II-3D	Computer Graphics II, 3D Image – Analysis and Synthesis	Prof. Dr. Hans-Peter Seidel, Dr. Marcus Magnor	6	LAG, LAB
Hauptstudium	CS 650 / MM	Multimedia	Prof. Dr. Philipp Slusallek	6	LAG, LAB
Hauptstudium	CS 650 / DIC	Differential Equations in Image Processing and Computer Vision	Prof. Dr. Joachim Weickert	9	LAG, LAB
Hauptstudium	CS 650 / DGAIP	Differential Geometric Aspects of Image Processing	Prof. Dr. Joachim Weickert	5	LAG, LAB
Hauptstudium	CS 750 / IIAM	Introduction to Image Acquisition Methods	Prof. Dr. Joachim Weickert	4	LAG, LAB
Hauptstudium	CS 750 / PMIP	Probabilistic Methods in Image Processing	Prof. Dr. Joachim Weickert	4	LAG, LAB
Hauptstudium	CS 750 / PR	Pattern Recognition	Prof. Dr. Joachim Weickert	5	LAG, LAB
Hauptstudium	EFS	Elementare schul- praktische Studien	StD Peter Mirold	7	LAG. LAB
Hauptstudium	VFS	Vertiefte schul- praktische Studien	StD Peter Mrold	9	LAG. LAB
Hauptstudium	Ddl	Didaktik der Informatik	StD Peter Mirold	9	LAG. LAB

Modul Ringvorlesungen Perspektiven der Informatik					CS 101
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1	1	jährlich	1 Semester	2	2

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

Dozent/inn/en Professoren der Fachrichtung

Zuordnung zum Curriculum 1. Semester, Wahlpflicht (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Keine

Leistungskontrollen / Prüfungen Positive Bewertung von mindestens drei schriftlichen
Zusammenfassungen verschiedener Vorträge

Lehrveranstaltungen / SWS Vorlesung / 2 SWS

Arbeitsaufwand 60 h = 30 h Präsenz- und 30 h Eigenstudium

Modulnote ---

Lernziele / Kompetenzen

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

Inhalt

Querschnitt durch die Forschungsthemen der Saarbrücker Informatik. Die Themen spannen einen attraktiven Bogen von aktuellster Forschung zu anspruchsvollen Problemen der industriellen Praxis.

Weitere Informationen

Unterrichtssprache: deutsch

Modul Sprachkurs IT - Englisch					CS 101
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1	1 - 6	jährlich	1 Semester	2	2 bzw. 3

Modulverantwortliche/r	Dr. Peter Tischer
Dozent/inn/en	Dr. Peter Tischer
Zuordnung zum Curriculum	Wahl (LAG,LAB)
Zulassungsvoraussetzungen	Studierende müssen einen Einstufungstest bestehen bzw. den erfolgreichen Besuch eines dem Unicert Niveaus III oder IV gleichwertigen Kurses nachweisen.
Leistungskontrollen / Prüfungen	Regelmäßige Teilnahme (mindestens 80% aller Veranstaltungen). Ein Referat, Abschlussklausur.
Lehrveranstaltungen / SWS	Seminar / 2 SWS
Arbeitsaufwand	60- 90 h = 30 h Präsenz- und 30 - 60 h Eigenstudium

Modulnote ---

Lernziele / Kompetenzen

Der Kurs soll Studierenden bei der Erweiterung ihres wissenschaftlichen und technischen Wortschatzes helfen. Hierbei werden spezielle Bedürfnisse und Interessen beachtet.

Inhalt

Der Kurs beschäftigt sich mit Fach-Englisch in den Bereichen IT und Ingenieurwissenschaften. Dabei ist nicht vorgesehen, systematisch Vokabellisten abzuarbeiten. Es werden vielmehr typische Texttypen wie z.B. Vorlesungen, Tabellen, Anleitungen, wissenschaftliche Aufsätze und Texte aus Lehrwerken betrachtet. Dabei werden auch für „Tech Talk“ typische grammatische Phänomene wie z.B. Passiv, Nominalisierung oder Konditionalsätze besprochen.

Weitere Informationen

Unterrichtssprache: englisch

Literatur: Aktuelle Liste und Zusammenstellung von Material zu Beginn des Kurses

Modul Mathematik für Informatiker 1					CS 110 / Mfi1
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1	1-3	jährlich	1 Semester	6	9

Modulverantwortliche/r

Prof. Prof. Dr. Joachim Weickert

Dozent/inn/en

Prof. Dr. Joachim Weickert, Prof. Dr. Frank-Olaf Schreyer, Prof. Dr. Wolfram Decker

Zuordnung zum Curriculum

1. Semester / Pflicht
(nur LAB mit einem allgemeinbildendem Fach, das nicht Mathematik ist)

Zulassungsvoraussetzungen

Keine

Leistungskontrollen / Prüfungen

- Teilnahme an den Übungen und Bearbeitung der wöchentlichen Übungsaufgaben (50 Prozent der Übungspunkte werden zur Klausurteilnahme benötigt)
- Bestehen der Abschlussklausur oder der Nachklausur
- Die Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.

Lehrveranstaltungen / SWS

Vorlesung: 4 SWS (ca. 250 Studierende)
Übung: 2 SWS
Übungsgruppen mit bis zu 20 Studierenden

Arbeitsaufwand

270 h = 90 h Präsenz- und 180 h Eigenstudium

Lernziele / Kompetenzen

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

Inhalt

Die Zahlen geben die Gesamtzahl der Doppelstunden an.

DISKRETE MATHEMATIK UND EINDIMENSIONALE ANALYSIS

A. Grundlagen der diskreten Mathematik (8)

1. Mengen (1)
2. Logik (1)
3. Beweisprinzipien, incl. vollst. Induktion (1)
4. Relationen (1)
5. Abbildungen (2)
 - injektiv, surjektiv, bijektiv
 - Mächtigkeit, Abzählbarkeit
 - Schubfachprinzip
6. Primzahlen und Teiler (1)
7. Modulare Arithmetik (1)

B. Eindimensionale Analysis (22)

B.1 Zahlen, Folgen und Reihen (8)

8. Axiomatik der reellen Zahlen, sup, inf (1)
9. Komplexe Zahlen (1)
10. Folgen (1 1/2)
11. Landau'sche Symbole (1/2)
12. Reihen: Konvergenzkriterien, absolute Kgz. (2)
13. Potenzreihen (1/2)
14. Zahlendarstellungen (1/2)
15. Binomialkoeffizienten und Binomialreihe (1)

B.2 Eindimensionale Differentialrechnung (8)

16. Stetigkeit (1)
17. Elementare Funktionen (1)
18. Differenzierbarkeit (1 1/2)
19. Mittelwertsätze und L'Hospital (1/2)
20. Satz von Taylor (1)
21. Lokale Extrema, Konvexität, Kurvendiskussion (2)
22. Numerische Differentiation (1)

B.3 Eindimensionale Integralrechnung (6)

23. Das bestimmte Integral (2)
24. Das unbestimmte Integral und die Stammfunktion (1)
25. Uneigentliche Integrale (1)
26. Numerische Verfahren zur Integration (1)
27. Kurven und Bogenlänge (1)

Weitere Informationen

Unterrichtssprache: deutsch

Literatur:

P. Hartmann: Mathematik für Informatiker. Vieweg, 2003.
M.P.H. Wolff, P. Hauck, W. Küchlin: Mathematik für Informatik
und BioInformatik. Springer, 2004.

Modul Mathematik für Informatiker 2					CS 210 / Mfi2
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
2	2-5	jährlich	1 Semester	6	9

Modulverantwortliche/r

Prof. Prof. Dr. Joachim Weickert

Dozent/inn/en

Prof. Dr. Joachim Weickert, Prof. Dr. Frank-Olaf Schreyer, Prof. Dr. Wolfram Decker

Zuordnung zum Curriculum

2. Semester / Pflicht
(nur LAB mit einem allgemeinbildendem Fach, das nicht Mathematik ist)

Zulassungsvoraussetzungen

Keine

Leistungskontrollen / Prüfungen

- Teilnahme an den Übungen und Bearbeitung der wöchentlichen Übungsaufgaben (50 Prozent der Übungspunkte werden zur Klausurteilnahme benötigt)
- Bestehen der Abschlussklausur oder der Nachklausur
- Die Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.

Lehrveranstaltungen / SWS

Vorlesung: 4 SWS (ca. 250 Studierende)
Übung: 2 SWS
Übungsgruppen mit bis zu 20 Studierenden

Arbeitsaufwand

270 h = 90 h Präsenz- und 180 h Eigenstudium

Lernziele / Kompetenzen

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

Inhalt

ALGEBRAISCHE STRUKTUREN UND LINEARE ALGEBRA

C. ALGEBRAISCHE STRUKTUREN (5)

- 29. Gruppen (2)
- 30. Ringe und Körper (1)
- 31. Polynomringe über allgemeinen Körpern (1/2)
- 32. Boole'sche Algebren (1/2)

D. LINEARE ALGEBRA (21)

- 33. Vektorräume (2)
 - Def., Bsp.,
 - lineare Abb.
 - Unterraum,
 - Erzeugnis, lineare Abhängigkeit, Basis, Austauschatz
- 34. Lineare Abb. (Bild, Kern) (1)
- 35. Matrixschreibweise für lineare Abbildungen (1 1/2)
 - Interpretation als lineare Abbildungen
 - Multiplikation durch Hintereinanderausführung
 - Ringstruktur
 - Inverses
- 36. Rang einer Matrix (1/2)
- 37. Gauss-Algorithmus für lineare Gleichungssysteme: (2)
 - Gaußelimination (1)
 - Lösungstheorie (1)
- 38. Iterative Verfahren für lineare Gleichungssysteme (1)
- 39. Determinanten (1)
- 40. Euklidische Vektorräume, Skalarprodukt (1)
- 41. Funktionalanalytische Verallgemeinerungen (1)
- 42. Orthogonalität (2)
- 43. Fourierreihen (1)
- 44. Orthogonale Matrizen (1)
- 45. Eigenwerte und Eigenvektoren (1)
- 46. Eigenwerte und Eigenvektoren symmetrischer Matrizen (1)
- 47. Quadratische Formen und positiv definite Matrizen (1)
- 48. Quadriken (1)
- 50. Matrixnormen und Eigenwertabschätzungen (1)
- 51. Numerische Berechnung von Eigenwerten und Eigenvektoren(1)

Weitere Informationen

Unterrichtssprache: deutsch

Literatur:

P. Hartmann: Mathematik für Informatiker. Vieweg, 2003.
M.P.H. Wolff, P. Hauck, W. Küchlin: Mathematik für Informatik
und BioInformatik. Springer, 2004.

Modul Programmierung 1					CS 120 / P1
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1	1-3	jährlich	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Gert Smolka
Dozent/inn/en	Prof. Dr. Gert Smolka, Prof. Dr. Andreas Podelski, Prof. Dr.-Ing. Holger Hermanns
Zuordnung zum Curriculum	1. Semester / Pflicht
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	<ul style="list-style-type: none"> • 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	Vorlesung: 4 SWS (ca. 250 Studierende) Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz- und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

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)

Weitere Informationen

Unterrichtssprache: deutsch

Literatur:

Skript zur Vorlesung; siehe auch Literaturliste vom WS 02/03:
<http://www.ps.uni-sb.de/courses/prog-ws02/literatur.html>

Modul Programmierung 2					CS 220 / P2
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
2	2 - 4	jährlich	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Hans-Peter Lenhof,
Dozent/inn/en	Prof. Dr. Hans-Peter Lenhof, Prof. Dr. Bernd Finkbeiner, Prof. Dr. Raimund Seidel
Zuordnung zum Curriculum	2. Semester/ Pflicht
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	Vorlesung: 2 SWS Übung: 4 SWS Davon 2 SWS Tutorien mit bis zu 80 Studierenden und 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Lehrveranstaltungen / SWS	Vorlesung: 4 SWS (ca. 250 Studierende) Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz- und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Die Studierenden lernen die Syntax und Semantik der Programmiersprache C++ und die Grundprinzipien der objektorientierten Programmierung kennen. Sie erwerben die Fähigkeit Funktionen, Algorithmen und Klassen in der Programmiersprache C++ zu entwerfen und zu implementieren. Sie lernen Hilfswerkzeuge, wie Debugger und Profiler, aufwandsgerecht einzusetzen. Sie erhalten erste Einblicke in die Grundzüge des Software-Entwurfs und der Dokumentation. Sie erwerben erste Erfahrungen in der Einbindung und Verwendung vorhandener Spezialbibliotheken.

Inhalt

C++ und objektorientierte Programmierung

- Deklarationen und Definitionen von Variablen
- Einfache Datentypen
- Ausdrücke
- Kontrollstrukturen und Schleifenanweisungen
- Unterprogramme, Module, Parameterübergabe
- Polymorphismus: Überladen von Funktionen
- Gültigkeitsbereiche und Sichtbarkeit von Variablen
- Eingabe in und Ausgabe von Dateien
- Zeiger/Pointer
- Strukturierte Datentypen
- Funktionstemplates
- Abstrakte Datentypen und Klassen
- Zugriffsrechte: public, private, protected
- Konstruktoren, Kopierkonstruktoren, Zuweisungsoperatoren, Destruktoren
- Klassentemplates
- Vererbung
- Polymorphismus: Überladen von Operatoren
- C++-Standardbibliothek: Container (vector, list, queue, dequeue, map), Iteratoren, Funktoren
- **Hilfswerkzeuge**
 - Softwarerekonstruktion (make)
 - Debugger
 - Profiler

Elementare Datenstrukturen

- Felder, Listen, Queues, Stacks
- Binäre Suchbäume, balancierte Bäume
- Heaps

Hashing

- Kollisionen und Chaining
- Uniformes Hashing
- Universelles Hashing
- Cuckoo Hashing

Sortieren

- Insertion-Sort
- Merge-Sort, Divide-and-Conquer
- Quick-Sort
- Bucket-Sort

Graphen und Graph-Algorithmen

- Gerichtete und ungerichtete Graphen
- Pfade, Rundgänge und Kreise
- Zusammenhangskomponenten
- BFS und DFS
- Topologisches Sortieren
- Starke Zusammenhangskomponenten

Weitere Informationen

Unterrichtssprache: deutsch

Literatur:

C++

- [Breymann.](#) *C++ Einführung und professionelle Programmierung*
- [C++-Referenz](#)
- Stanley B. Lippman, Josee Lajoie. *C++ Primer*. Addison Wesley 1998. (ISBN: 0201824701)
- Bjarne Stroustrup. *The C++ Programming Language*. Addison Wesley Publishing Company 1997. (ISBN: 0201889544)

Algorithmen und Datenstrukturen

- T.H.Cormen, C.E.Leiserson, R.L.Rivest and C. Stein. *Introduction to Algorithms - Second Edition*. McGraw-Hill, 2001. (ISBN: 0262531968)
- K.Mehlhorn and S.Näher. *The LEDA Platform of Combinatorial and Geometric Computing*. Cambridge University Press, 1999. (ISBN 0521563291)
- K.Mehlhorn, P.Sanders. [Informatik V: Data Structures and Algorithms](#). Skript zur Vorlesung im SS 2000

Empfohlene Vorkenntnisse:

- Programmierung 1
- Mathematik für Informatiker 1 und Mathematikveranstaltungen im Studiensemester oder vergleichbare Kenntnisse aus sonstigen Mathematikveranstaltungen

Modul Systemarchitektur					CS 230 / SysArch
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
2	2 -4	jährlich	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. W.-J. Paul

Dozent/inn/en Prof. Dr. W.-J. Paul

Zuordnung zum Curriculum 2. Semester / Pflicht

Zulassungsvoraussetzungen Keine

Leistungskontrollen / Prüfungen Studienleistungen:
die Vorlesungen hören, nach bearbeiten und gegebenenfalls verstehen; die Übungen allein oder in Gruppen bearbeiten; erfolgreich bearbeitete Übungen in der Übungsgruppe vortragen und gegebenenfalls erklären können.

Prüfungsleistungen:
erfolgreiche Bearbeitung von 50 % der Übungsaufgaben berechtigt zur Teilnahme an den Klausuren (2 Klausuren, 1 Nachklausur)

Die Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.

Lehrveranstaltungen / SWS Vorlesung: 4 SWS (ca. 250 Studierende)
Übung: 2 SWS
Übungsgruppen mit bis zu 20 Studierenden

Arbeitsaufwand 270 h = 90 h Präsenz- und 180 h Eigenstudium

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

Lernziele / Kompetenzen

Die Studierenden sollen lernen, ein einfaches System, bestehend aus Prozessor, Compiler und Betriebssystemkern

- zu spezifizieren
- zu designen und

per Korrektheitsbeweis (weil das am schnellsten geht) zu erklären, warum es funktioniert

Inhalt

1. Hardware
 - a. Boole'sche Algebra und Schaltkreise
 - b. Elementare Rechnerarithmetik
 - c. ALU (Konstruktion und Korrektheit)
 - d. Sequentieller vereinfachter DLX-Prozessor (Konstruktion und Korrektheit)
2. Compiler für eine C-ähnliche Sprache
 - a. Syntax
 - b. Semantik
 - c. Korrektheitskriterium für die Übersetzung
 - d. Code-Generierung
 - e. Ausgewählte Teile des Korrektheitsbeweises
3. Betriebssystemkern
 - a. Interrupt-Mechanismus
 - b. Memory-Management-Unit
 - c. Spezifikation des Verhaltens (CVM)
 - d. Konstruktion
 - e. Ausgewählte Teile des Korrektheitsbeweises

Empfohlene Vorkenntnisse:

Programmierung 1 und Mathematik für Informatiker 1 oder vergleichbare Veranstaltungen der Mathematik

Modul Grundzüge von Algorithmen und Datenstrukturen					CS 340 / GrADS
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
3	3 - 5	jährlich	1 Semester	4	6

Modulverantwortliche/r	Prof. Dr. Raimund Seidel,
Dozent/inn/en	Prof. Dr. Markus Bläser, Prof. Dr. Kurt Mehlhorn, Prof. Dr. Raimund Seidel
Zuordnung zum Curriculum	3. Semester / Pflicht
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	<p>Studienleistungen: die Vorlesungen hören, nach bearbeiten und gegebenenfalls verstehen; die Übungen allein oder in Gruppen bearbeiten; erfolgreich bearbeitete Übungen in der Übungsgruppe vortragen und gegebenenfalls erklären können.</p> <p>Prüfungsleistungen: erfolgreiche Bearbeitung von 50 % der Übungsaufgaben berechtigt zur Teilnahme an den Klausuren (2 Klausuren, 1 Nachklausur)</p> <p>Die Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.</p>
Lehrveranstaltungen / SWS	Vorlesung: 2 SWS Übung: 2 SWS in Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	180 h = 60 h Präsenz- und 120 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.
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.

Weitere Informationen

Unterrichtssprache: deutsch

Empfohlene Vorkenntnisse:

- Programmierung 1 und 2
- Mathematik für Informatiker 1 und 2
oder vergleichbare Veranstaltungen der Mathematik

Modul Informationssysteme					CS 330 / InfoSys
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
4	4 - 6	jährlich	1 Semester	4	6

Modulverantwortliche/r

Prof. Dr. Gerhard Weikum

Dozent/inn/en

Prof. Dr. Gerhard Weikum, Prof. Dr. Christoph Koch

Zuordnung zum Curriculum

4. Semester / Pflicht

Zulassungsvoraussetzungen

Keine

Leistungskontrollen / Prüfungen

Es werden 6 benotete Leistungspunkte vergeben, wenn folgende Voraussetzungen erfüllt sind:

1. erfolgreiche Teilnahme an zwei Teilklausuren in der Mitte und am Ende des Semesters
oder
erfolgreiche Teilnahme an einer Teilklausur und der Nachklausur Anfang Oktober
2. erfolgreiche Teilnahme an den Übungen:
Abgabe eines kleinen Programmierprojekts und Erreichen von mehr als der Hälfte der möglichen Punkte bei kurzen Multiple-Choice-Tests in den Übungsstunden.

Die Note wird aus den Ergebnissen der zwei bestandenen (Teil-) Klausuren berechnet.

Lehrveranstaltungen / SWS

Vorlesung: 3 SWS
Übung: 1 SWS
Übungsgruppen mit bis zu 20 Studierenden

Arbeitsaufwand

180 h = 60 h Präsenz- und 120 h Eigenstudium

Modulnote

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

Lernziele / Kompetenzen

Die Vorlesung vermittelt grundlegende Kenntnisse über Konzepte und Schnittstellen von Datenbanksystemen und anderen Arten von Informationsdienstsoftware sowie der Anwendungsentwicklungswerkzeuge zur Realisierung von Informationssystemen. Besonderes Augenmerk wird auf die logische Ebene des ANSI 3-Schichtenmodells gelegt.

Inhalt

Schwerpunktt Themen sind das relationale Modell, Anfragesprachen für Datenbanksysteme, Nichtausdrückbarkeitsbeweise, Datenmodellierung, Designtheorie und Normalformen fuer relationale Schemata, Aequivalenz und Minimierung von Anfragen, Integritaetsbedingungen, Datenintegration und aktuelle Themen wie Webinformationssysteme, Information Retrieval, und die Handhabung von unvollstaendiger Information.

Die notwendigen Grundlagen werden in der Vorlesung eingeführt.

Weitere Informationen

Unterrichtssprache: deutsch

Literatur:

- Alfons Kemper, Andre Eickler: Datenbanksysteme - eine Einführung, Oldenbourg, 2001
- Serge Abiteboul, Richard Hull, Victor Vianu: Foundations of Databases, Addison-Wesley, 1995
- Jiawei Han, Micheline Kamber: Data Mining - Concepts and Techniques, Morgan Kaufmann, 2001

Empfohlene Vorkenntnisse:

Programmierung 1 & 2, Softwarepraktikum, Theoretische Informatik

Modul Nebenläufige Programmierung					CS 430
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
4	4 - 6	jährlich	1 Semester	4	6

Modulverantwortliche/r

Prof. Dr. Ing. Holger Hermanns

Dozent/inn/en

Prof. Dr.-Ing. Holger Hermanns,
Prof. Dr. Gert Smolka,
Prof. Bernd Finkbeiner, PhD

Zuordnung zum Curriculum

4. Semester/ Pflicht

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 Wochen vor Vorlesungsbeginn des Folgesemesters statt.

Lehrveranstaltungen / SWS

Vorlesung: 2 SWS
Übung: 2 SWS
Übungsgruppen mit bis zu 20 Studierenden

Arbeitsaufwand

180 h = 60 h Präsenz- und 120 h Eigenstudium

Modulnote

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

Lernziele / Kompetenzen

Die Studierenden

- begreifen Nebenläufigkeit als Konzept
- kennen Nebenläufigkeit in der Praxis
- kennen die theoretischen Grundlagen von Nebenläufigkeit
- werden mit den Schwierigkeiten von Nebenläufigkeit vertraut
- verstehen den Kalkuel kommunizierender Prozesse (CCS)
- lernen das Programmieren von Nebenläufigkeit und werden mit Verfahren zur Analyse und Programmierunterstützung vertraut

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-2-peer, Datenbanken, Internet)

Die Schwierigkeit von Nebenläufigkeit

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

Grundlagen der Nebenläufigkeit

- Sequentielle 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
- Interaktion
- Strukturelle operationelle Semantik
- Gleichheit von Beobachtungen
- Implementierungsrelationen
- CCS mit Datentransfer

Programmieren von Nebenläufigkeit

- Java vs. C++
- Objekte in Java
- Sockets, Protokolle, Datenströme in Java
- Shared Objects und Threads in Java
- Shared Objects und Threads als Transitionssysteme
- Monitore und Semaphoren

Analyse und Programmierunterstützung

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

Weitere Informationen

Unterrichtssprache: deutsch

Literatur:

-- Skript zur Vorlesung

-- Jeff Magee and Jeff Kramer.

Concurrency: State Models & Java Programs
(2nd Edition), Wiley, 2006

<http://www-dse.doc.ic.ac.uk/concurrency/>

-- Robin Milner.

Communication and Concurrency
Prentice Hall, 1995

Modul Grundzüge der Theoretischen Informatik					CS CS 420 / TheoInf
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
3	3 - 5	jährlich	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Raimund Seidel
Dozent/inn/en	Prof. Dr. Bernd Finkbeiner, Prof. Dr. Kurt Mehlhorn, Prof. Dr. W.J. Paul, Prof. Dr. Raimund Seidel, Prof. Dr. Reinhard Wilhelm
Zuordnung zum Curriculum	3. Semester / Pflicht
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	<ul style="list-style-type: none"> • Erfolgreiche Bearbeitung der Übungsaufgaben; • Bestehen von Zwischentests und einer Abschlussklausur • Die Note ergibt sich aus den Tests und dem Klausurergebnis. • Eine Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.
Lehrveranstaltungen / SWS	Vorlesung 4 SWS Übung 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz- und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

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

Weitere Informationen

Unterrichtssprache: deutsch

Literatur:

- Ingo Wegener: Theoretische Informatik - eine algorithmenorientierte Einführung.
- Harry R. Lewis, Christos H. Papadimitriou: Elements of the Theory of Computation
- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman: Introduction to Automata Theory, Languages, and Computation
- Uwe Schöning: Theoretische Informatik - kurzgefasst
- Michael Sipser: Introduction to the Theory of Computation
- Norbert Blum: Theoretische Informatik

Empfohlene Vorkenntnisse:

- Programmierung 1, Programmierung 2
- Mathematik für Informatiker 1 und 2 oder vergleichbare Veranstaltungen der Mathematik

Modul Softwaredesignpraktikum					CS CS 320 / SoDePra
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
3	3 - 5	Jährlich in den Sommerferien	6 Wochen	8	9

Modulverantwortliche/r	Prof. Dr. Andreas Zeller
Dozent/inn/en	Prof. Dr. Andreas Zeller, Prof. Dr. Philipp Slusallek, Prof. Dr. Holger Hermanns
Zuordnung zum Curriculum	3. Semester / Pflicht

Zulassungsvoraussetzungen Keine

- Leistungskontrollen / Prüfungen**
- Individuelles Bearbeiten einer komplexen Entwurfs- und Programmieraufgabe
 - Erfolgreiches Erstellen im Team eines komplexen Software-Produkts, insbesondere
 - Einreichen der erforderlichen Dokumente
 - Abnahme des Endprodukts durch den Kunden
 - Einhaltung der Termin- und Qualitätsstandards

- Das Praktikum findet als 6-wöchiges Blockpraktikum in der Sommerpause statt. Dem eigentlichen Praktikum vorangestellt ist eine im Selbststudium zu erarbeitende größere Programmieraufgabe. In dieser erarbeitet sich der Studierende hinreichende praktische Programmierkenntnisse in C/C++. Die erfolgreiche Anfertigung der Programmieraufgabe wird vor Beginn des eigentlichen Praktikums überprüft.

- In den ersten zwei Wochen des Blockpraktikums wird dann auf der Basis eines fertig ausgearbeiteten Pflichtenheftes - mit vollständig automatisch testbaren Anforderungen - das System entworfen. Nach einer Woche muss das UML-Objektmodell (Klassen und ihre Beziehungen) stehen. Meilenstein nach einer weiteren Woche ist der vollständige Entwurf, der neben einem überarbeiteten Objektmodell eine Reihe von Standard-Szenarien mit Sequenzdiagrammen und Unit-Tests beschreibt.

- In der Entwurfsphase beurteilt ein Dozent oder ein wissenschaftlicher Mitarbeiter jeden Entwurf in zwei Kolloquien je Gruppe und Tutor. Hierbei geht der Prüfer den Entwurf mit der Gruppe durch, prüft, ob die gesamte Gruppe mit dem Entwurf vertraut ist, und gibt eine Reihe von Auflagen mit auf den Weg, um den Entwurf zu verbessern.

- Nach abgeschlossenem Entwurf wird ein überarbeitetes Pflichtenheft ausgegeben, in dem sich einige Details geändert haben. Die Studierenden wissen zu Beginn, dass sich Details ändern können (aber nicht, welche Details das sind) und streben angesichts dieses Risikos einen möglichst flexiblen Entwurf an.

- In den folgenden zwei Wochen wird das System implementiert. Meilenstein ist hier das Bestehen eines vorgegebenen automatischen Tests, der die gesamte Funktionalität des Systems abdeckt.

- In den letzten zwei Wochen wird das System eingesetzt. Anstelle der Kundenorientierung tritt dabei ein kompetitives Element, da sich jede Implementierung für ein Turnier qualifizieren muss, und anschliessend in dem Turnier möglichst erfolgreich bestehen soll. Hier können insbesondere fortgeschrittene algorithmische Methoden der Informatik ihren Nutzen zeigen.

Lehrveranstaltungen / SWS

Vorlesung 2 SWS
Praktikum 6 SWS
Teamarbeit in Gruppen bis zu 6 Studierenden

Arbeitsaufwand

150h vorbereitendes Selbststudium & Programmierung

Im 6-wöchigem Block (30 Arbeitstage):

- 12 Vorlesungen a 2h
- Täglich 1h Besprechung mit Tutor
- Täglich 4h Arbeit im Team
- Täglich 3h Selbststudium & Programmierung

gesamt: 414h

Modulnote

Wird aus in mündlichen Prüfungen und in der Projektarbeit erbrachten Leistungen ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Die Studierenden erwerben die Fähigkeit, im Team zu arbeiten und Probleme der Informatik zu lösen.

Die Studierenden wissen, welche Probleme beim Durchführen eines Software-Projekts auftreten können, und wie man damit umgeht.

Sie können eine komplexe Aufgabenstellung eigenständig in ein Software-Produkt umsetzen, das den Anforderungen des Kunden entspricht. Hierfür wählen sie einen passenden Entwicklungsprozess, der Risiken frühzeitig erkennt und minimiert, und wenden diesen an. Sie sind vertraut mit Grundzügen des Software-Entwurfs wie schwache Kopplung, hohe Kohäsion, Geheimnisprinzip sowie Entwurfs- und Architekturmustern und sind in der Lage, einen Entwurf anhand dieser Kriterien zu erstellen, zu beurteilen und zu verbessern.

Sie beherrschen Techniken der Qualitätssicherung wie Testen und Gegenlesen und wenden diese an.

Sie erlernen im praktischen Umgang diverse Facetten des Projektmanagements, der Teamarbeit und Selbstorganisation, sowie Zeitmanagement, Kundenorientierung, und Kommunikationskompetenz.

Inhalt

Software-Entwurf (objektorientierter Entwurf mit UML)
Software-Prozesse (Wasserfall, inkrementelles Modell, agile Modelle)
Arbeiten im Team
Projektplanung und -Durchführung
Qualitätssicherung
Programmierwerkzeuge (Versionskontrolle, Konstruktion, Test, Fehlersuche)

Weitere Informationen

Unterrichtssprache: deutsch

Literatur:

Balzert, Einführung in die Softwaretechnik I + II
Gamma et al., Entwurfsmuster

Empfehlung:

Eine Teilnahme am Praktikum ohne vorherige erfolgreiche Teilnahme an den Modulen Programmieren 1 und Programmieren 2 ist nicht sinnvoll.

Modul Seminar					CS 500
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5-9	5 - 9	Jedes Semester	1 Semester	3	7

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

Dozent/inn/en Professoren der Fachrichtung

Zuordnung zum Curriculum Pflicht / Wahl

Zulassungsvoraussetzungen Keine.

Leistungskontrollen / Prüfungen

- Beiträge zur Diskussion
- Thematischer Vortrag
- Schriftliche Ausarbeitung
- Mündliche Abschlussprüfung über das gesamte Themengebiet

Lehrveranstaltungen / SWS Seminar 3 SWS (bis zu 25 Studierende)

Arbeitsaufwand 210h = 45 h Präsenz und 165 h Eigenstudium

Modulnote Die Modalitäten der Notenvergabe werden vom verantwortlichen Hochschullehrer festgelegt.

Lernziele / Kompetenzen

Die Studierenden haben am Ende der Veranstaltung ein tiefes Verständnis aktueller oder fundamentaler Aspekte eines spezifischen Teilbereiches der Informatik erlangt.

Sie haben Kompetenz im eigenständigen wissenschaftlichen Recherchieren, Einordnen, Zusammenfassen, Diskutieren, Kritisieren und Präsentieren von wissenschaftlichen Erkenntnissen gewonnen.

Inhalt

Praktisches Einüben von

- reflektierender wissenschaftlicher Arbeit,
- Analyse und Bewertung wissenschaftlicher Aufsätze,
- Verfassen eigener wissenschaftlicher Zusammenfassungen
- Diskussion der Arbeiten in der Gruppe
- Erarbeiten gemeinsamer Standards für wissenschaftliche Arbeiten
- Präsentationstechnik

Spezifische Vertiefung in Bezug auf das individuelle Thema des Seminars.

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
- Mündliche Prüfung über das erarbeitete Themengebiet

Weitere Informationen

Literatur:

dem Thema entsprechend

Modul Operating Systems, Core Course					CS 551 / OS
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mind. alle 2 Jahre	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Helge Scheidig

Dozent/inn/en Prof. Dr. Helge Scheidig

Zuordnung zum Curriculum Wahl- bzw. Wahlpflichtbereich
Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Keine

Leistungskontrollen / Prüfungen Regular attendance of classes and tutorials
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 Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Arbeitsaufwand 270 h = 90 h of classes and 180 h private study

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

Lernziele / Kompetenzen

Knowledge of fundamental characteristics of operating systems

Inhalt

- development
- Tasks and organisation forms
- Processes: general characteristics, synchronization
- Processes: implementational characteristics,
- interruption systems
- operating equipment
- scheduling
- main storage maintenance
- message exchange, procedural and message oriented operating systems,
- structuring operating systems
- modelling asynchronous processes through petri-networks,
- dealing with deadlocks, security
- coordination in systems for computer aided group work,
- coordination of multimedia streams in multimedia systems

Weitere Informationen

Unterrichtssprache: deutsch

Literatur:

Updated list at the beginning of each semester

A. Tanenbaum: Operating Systems (Design und Implementation), Prentice Hall, 1995)

Modul Computer Graphics, Core Course					CS 552 / CG
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mind. alle 2 Jahre	1 Semester	6	9

Modulverantwortliche/r

Prof. Dr. Philipp Slusallek

Dozent/inn/en

Prof. Dr. Philipp Slusallek, Prof. Dr. Hans-Peter Seidel,
Dr. Marcus Magnor

Zuordnung zum Curriculum

Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen

Englischkenntnisse gemäß § 5 des Fachspezifischen
Anhangs zur Studienordnung und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

- Successful completion of at least 50% of the exercises
- Successful participation in rendering competition
- Final written exam

Final grade determined by result of the exam and the
rendering competition

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

Lehrveranstaltungen / SWS

Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Arbeitsaufwand

270 h = 90 h of classes and 180 h private study

Modulnote

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

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 focuses on image synthesis or rendering. After introducing of physical background and the representations used in graphics it discusses the two basic algorithms for image synthesis: ray tracing and rasterization. In this context we present related topics like texturing, shading, aliasing, sampling, and many more. As part of the practical exercises the students incrementally build their own ray tracing system or hardware-based visualization application. A final rendering competition allows students to implement their favorite advanced algorithm and use it in a high-quality rendering.

Inhalt

- Fundamentals of digital image synthesis
 - Physical laws of light transport
 - Human visual system and perception
 - Colors and Tone-Mapping
 - Signal processing and anti-aliasing
 - Materials and reflection models
 - Geometric modeling
 - Camera models
- Ray Tracing
 - Recursive ray tracing algorithm
 - Spatial index structures
 - Sampling approaches
 - Parallel and distributed algorithms
- Rasterization and Graphics Hardware
 - Homogeneous coordinates, transformations
 - Hardware architectures
 - Rendering pipeline
 - Shader programming and languages
 - OpenGL

Weitere Informationen

Unterrichtssprache: English

Literatur:

- Alan Watt, 3D Computer Graphics, Addison-Wesley, 1999
- James Foley, Andries Van Dam, et al., Computer Graphics : Principles and Practice, 2. Edition, Addison-Wesley, 1995
- Andrew Glassner, Principles of Digital Image Synthesis, 2 Volumes, Morgan Kaufman, 1996
- Peter Shirley, Realistic Ray-Tracing, AK Peters
- Andrew Woo, et al., OpenGL Programming Guide, 3. Edition, Addison-Wesley, 1999
- Randima Fernando, GPU Gems, Addison-Wesley, 2004

Modul Database Systems, Core Course					CS 553 / DBS
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mind. alle 2 Jahre	1 Semester	6	9

Modulverantwortliche/r

Prof. Dr. Christoph Koch

Dozent/inn/en

Prof. Dr. Christoph Koch, Prof. Dr. Gerhard Weikum

Zuordnung zum Curriculum

Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen

Keine

Leistungskontrollen / Prüfungen

- Passing a two-hour written exam at the end of the semester
- Successful demonstration of programming project (teams of 2 students are allowed)

Grades are based on written exam (100 points); successful demonstration of the programming project is a requirement for the admission to the exam. It is possible to obtain up to ca. 20 bonus points for the programming project (for efficient implementations and the implementation of advanced query optimization techniques)

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

Lehrveranstaltungen / SWS

Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Arbeitsaufwand

270 h = 90 h of classes and 180 h private study

Modulnote

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

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, a DBMS kernel will be implemented and its performance evaluated. The goal of this implementation project is to work with the techniques introduced in the lectures and to understand them and their practical implications to a depth that would not be attainable by purely theoretical study. Moreover, an important goal of this project - and the course as a whole - is to communicate the essential difference between being a mere programmer and being a systems expert: The techniques taught in the course should allow the participant, starting the implementation project with a naive prototype, to attain query processing performance improvements of many orders of magnitude, far beyond what could be achieved by good programming alone.

Inhalt

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

- physical storage; disks, pages, records, clustering
- tree- and hash-indexes
- query processing: sorting on disk, pipelined evaluation, nested-loop-, hash- and merge-joins, ...
- query optimization (algebraic query rewriting, join reordering, selectivity estimations, histograms and cost-based optimization)
- database tuning
- transactions; concurrency control and recovery
- distributed databases: vertical and horizontal partitioning, distributed query evaluation and optimization, distributed transaction management
- (two-phase commit, ...), redundancy
- XML-, object-oriented-, and object-relational databases

Weitere Informationen

Unterrichtssprache: English

Literatur:

Ramakrishnan and Gehrke, Database Management Systems, 3rd Edition, McGraw-Hill 2002 (ISBN 0-07-115110-9) -- English.

or

Kemper/Eickler, "Datenbanksysteme", 5th edition, Oldenbourg Verlag -- German

Modul Data Networks, Core Course					CS 554 /DN
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mind. alle 2 Jahre	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Holger Hermanns

Dozent/inn/en Prof. Dr. Holger Hermanns

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

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

Lecture 4 h (weekly)
 Tutorial 2 h (weekly)
 Tutorials in groups of up to 20 students

Arbeitsaufwand 270 h = 90 h of classes and 180 h private study

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

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
- Protokols 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

Weitere Informationen

Unterrichtssprache: English

Literatur:

- James F. Kurose, Keith W. Ross: Computer networking : a top-down approach featuring the internet. – 3rd ed. – Boston : Addison-Wesley, 2004.
- Dimitri Bertsekas, Robert Gallager: Data networks. – 2nd ed. – Saddle River, NJ : Prentice Hall, 1992.
- Gerald J. Holzmann: Design and Validation of Computer Protocols. – Englewood Cliffs, NJ : Prentice–Hall, 1991.
- Andrew S. Tanenbaum: Computer networks. – 4th ed. – Upper Saddle River, NJ : Pearson Education, 2003.

Modul Information Retrieval and Data Mining, Core Course					CS 555 / IRDM
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mind. alle 2 Jahre	1 Semester	6	9

Modulverantwortliche/r

Prof. Dr. Gerhard Weikum

Dozent/inn/en

Prof. Dr. Gerhard Weikum

Zuordnung zum Curriculum

Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen

Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Passing 2 of 3 written exams (midterm, final and re-exam)
- Presentation of a solution during a tutorial (at least once)
- For each additional presentation up to 3 bonus points can be gained
- Passing the practical exercises (teams of up to two students)
- Up to 3 bonus points can be gained for the overall quality of the solutions

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

Lehrveranstaltungen / SWS

Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Arbeitsaufwand

270 h = 90 h of classes and 180 h private study

Modulnote

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

Lernziele / Kompetenzen

The lecture teaches mathematical models and algorithms that form the basis for search engines for the Web, intranets, and digital libraries and for data mining and analysis tools.

Inhalt

Information Retrieval and Data Mining are technologies for searching, analyzing and automatically organizing text documents, multi-media documents, and structured or semistructured data. The course teaches mathematical models and algorithms that form the basis for search engines for the Web, intranets, and digital libraries and for data mining and analysis tools. The fundamentals are models and methods from linear algebra and regression (e.g. singular-value decomposition) as well as probability theory and statistics (e.g. Bayesian networks and Markov chains). The exercises include practical tasks for the implementation of a simple search engine in Java.

Weitere Informationen

Unterrichtssprache: English

Literatur:

Information Retrieval

- *C.D. Manning, H. Schütze: Foundations of Statistical Natural Language Processing*, MIT Press, 1999
- *S. Chakrabarti: Mining the Web: Analysis of Hypertext and Semistructured Data*, Morgan Kaufmann, 2002
- *R. Baeza-Yates, B. Ribeiro-Neto: Modern Information Retrieval*, Addison-Wesley, 1999.
- *N. Fuhr: Information Retrieval*, Skriptum zur Vorlesung im SS 2002, Uni Dortmund.

Data Mining

- *J. Han, M. Kamber: Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2000
- *R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification*, John Wiley & Sons, 2001

Java

- *Go To Java 2*
- *Thinking in Java*

Modul Artificial Intelligence, Core Course					CS 556 / AI
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mind. alle 2 Jahre	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Wolfgang Wahlster
Dozent/inn/en	Prof. Dr. Wolfgang Wahlster
Zuordnung zum Curriculum	Wahl (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung
Leistungskontrollen / Prüfungen	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials • Solving of weekly assignments • Passing the final written exam <p>A re-exam takes place during the last two weeks before the start of lectures in the following semester.</p>
Lehrveranstaltungen / SWS	<p>Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students</p>
Arbeitsaufwand	270 h = 90 h of classes and 180 h private study
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.
Lernziele / Kompetenzen	Knowledge about the fundamentals of artificial intelligence

Inhalt

Problem-solving:

- Uninformed- and informed search procedures
- Adversarial search
- Knowledge and reasoning:
- First-order logic, Inference in first-order logic
- Knowledge representation

Planning:

- Planning
- Planning and acting in the real world

Uncertain knowledge and reasoning:

- Uncertainty
- Probabilistic reasoning
- Simple & complex decisions

Learning:

- Learning from observations
- Knowledge in learning
- Statistical learning methods
- Reinforcement learning

Communicating, perceiving, and acting:

- Communication
- Natural language processing
- Perception

Weitere Informationen

Unterrichtssprache: English

Literatur:

An updated list of used literature will be issued at the beginning of the semester.

- S. Russell, P. Norvig: Artificial Intelligence – A Modern Approach (2nd Edition), Prentice Hall Series in AI

Modul Computer Architecture, Core Course					CS 558 / CAR
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mind. alle 2 Jahre	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. W.-J. Paul
Dozent/inn/en	Prof. Dr. W.-J. Paul
Zuordnung zum Curriculum	Wahl (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	For graduate students: Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung
Leistungskontrollen / Prüfungen	<p>Studying: Students should listen to the lectures, read the lecture notes afterwards and understand them. They should solve the exercises alone or in groups. Students must present and explain their solutions during the tutorials.</p> <p>Exams: Students who have solved 50 % of all exercises are allowed to participate in an oral exam at the end of the semester.</p>
Lehrveranstaltungen / SWS	<p>Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students</p>
Arbeitsaufwand	270 h = 90 h of classes and 180 h private study
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.
Lernziele / Kompetenzen	<p>After attending this lecture students know how to design pipelined processors with interrupt mechanisms, caches and MMUs. Given a benchmark they know how to analyse, whether a change makes the processor more or less cost effective.</p>

Inhalt

General comment: constructions are usually presented together with correctness proofs

- Complexity of Architectures
 - Hardware cost and cycle time
 - Compilers and benchmarks
- Circuits
 - Elementary computer arithmetic
 - Fast adders
 - Fast multipliers
- Sequential processor design
 - DLX instruction set
 - Processor design
- Pipelining
 - Elementary pipelining
 - Forwarding
 - Hardware-Interlock
- Interrupt mechanisms
 - Extension of the instruction set
 - Interrupt service routines
 - hardware construction
- Caches
 - Specification including consistency between instruction and data cache
 - Cache policies
 - Bus protocol
 - Hardware construction (k-way set associative cache, LRU replacement, realisation of bus protocols by automat)
- Operating System Support
 - Virtual and Physical machines
 - Address translation
 - Memory management unit (MMU) construction
- Virtual memory simulation

Weitere Informationen

Unterrichtssprache: English

Literatur:

An updated list of used literature will be issued at the beginning of the semester.

Müller-Paul: Computer Architecture: Correctness and Complexity; Springer 2000

Current scientific papers

Modul Security, Core Course					CS 559 / SEC
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mind. alle 2 Jahre	1 Semester	6	9

Modulverantwortliche/r

PD Dr. Werner Stephan

Dozent/inn/en

PD Dr. Werner Stephan, Dr. Hutter

Zuordnung zum Curriculum

Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen

Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

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

Lecture 4 h (weekly)
 Tutorial 2 h (weekly)
 Tutorials in groups of up to 20 students

Arbeitsaufwand

270 h = 90 h of classes and 180 h private study

Modulnote

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

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

Weitere Informationen

Unterrichtssprache: English

Literatur:

Matt Bishop: Computer Security, Addison-Weseley, 2003
Matt Bishop: Introduction in Computer Security. Addison-Weseley, 2003
Johannes Buchmann: Einführung in die Kryptographie, Springer, 2001
Ross Anderson: Security Engineering. Wiley & Sons, 2001
Claudia Eckert: IT-Sicherheit. Oldenbourg, 2001

Modul Software Engineering, Core Course					CS 560 / SE
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mind. alle 2 Jahre	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Andreas Zeller

Dozent/inn/en Prof. Dr. Andreas Zeller

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

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

Lecture 4 h (weekly)
 Tutorial 2 h (weekly)
 Tutorials in groups of up to 20 students

Arbeitsaufwand 270 h = 90 h of classes and 180 h private study

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

Lernziele / Kompetenzen

The students know and apply modern software development techniques

They are aware of advanced quality assurance techniques such as test coverage, program analysis, and verification and know about the appropriate standards.

They know modern paradigms of programming and design, and know when to use them.

They know the standards of project management and project organization and can assess the state of given projects as well as suggest consequences to reach specific targets.

Inhalt

- Software Processes (Testing process, ISO 9000, maturity model, extreme programming)
- Modeling and design (requirements engineering, formal specification, proofs, model checking)
- Programming paradigms (aspect-oriented, generative, and component-based programming)
- Validation (Testing, Reliability assessment, tools)
- Software maintenance (configuration management, reengineering, restructuring)
- Project skills (organization, structure, estimations)
- Human resources (communication, assessment)
- Controlling (metrics, change requests, risk and quality management)

Weitere Informationen

Unterrichtssprache: English

Literatur:

Balzert, Softwaretechnik I and II
Own lecture notes

Modul Compiler Construction, Core Course					CS 561 / CC
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mind. alle 2 Jahre	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Reinhard Wilhelm

Dozent/inn/en Prof. Dr. Reinhard Wilhelm

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

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

Lecture 4 h (weekly)
 Tutorial 2 h (weekly)
 Tutorials in groups of up to 20 students

Arbeitsaufwand 270 h = 90 h of classes and 180 h private study

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

Lernziele / Kompetenzen

The students learn, how a source program is lexically, syntactically, and semantically analyzed, and how they re 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. Generative methods for compilation subtasks.

Weitere Informationen

Unterrichtssprache: English

Literatur:

R. Wilhelm, D. Maurer: Übersetzerbau, Springer 1997
 R. Wilhelm, D. Maurer: Compiler Design, Addison Wesley, 1995
 S. Muchnick: Advanced Compiler Design and Implementation, Morgan Kaufman 1997

Modul Automated Reasoning, Core Course					CS 571 / AR
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mind. alle 2 Jahre	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Bernd Finkbeiner, Prof. Dr. Jörg Siekmann

Dozent/inn/en Prof. Dr. Bernd Finkbeiner, Prof. Dr. Jörg Siekmann

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorials
- Weekly assignments
- Practical work with systems
- 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
 Lecture 4 h (weekly)
 Tutorial 2 h (weekly)
 Tutorials in groups of up to 20 students

Arbeitsaufwand 270 h = 90 h of classes and 180 h private study

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

Lernziele / Kompetenzen

The goal of this course is to provide familiarity with the main support systems for formal methods and computer supported mathematical reasoning, such as ISABELLE, HOL, OMEGA, TPS or COQ.

Inhalt

First and higher order calculi for automated reasoning systems
 Architecture of a first order theorem prover and system integration
 Higher Order Theorem Proving
 Decision procedures and SAF solver
 Proof Planning
 Proof objects and proof presentation
 Hands-on practical experience with the major systems for formal reasoning.

Weitere Informationen

Unterrichtssprache: English

Literatur:

Fitting, Melvin; First order Logic and Automated Theorem Proving. 2nd Edition, Springer, 1996.

Andrews, Peter; An Introduction to Mathematical Logic and Type theory: To Truth Through Proof. 2nd Edition, Kluwer Academic Publishers, 2002.

Books and Papers to ISABELLE, HOL; OMEGA, COQ and others.

Modul Theoretische Stammvorlesung Image Processing and Computer Vision					CS 572
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mindestens 1 in 2 Jahren	1 Semester	6	9

Modulverantwortliche/r

Prof. Dr. Joachim Weickert

Dozent/inn/en

Prof. Dr. Joachim Weickert

Zuordnung zum Curriculum

7. –9. Semester
Wahlpflicht / Theoretische Stammvorlesung
(LAG, LAB 1. Fach)

Zulassungsvoraussetzungen

Keine

Leistungskontrollen / Prüfungen

- Teilnahme an den Übungen und Bearbeitung der wöchentlichen Übungsaufgaben (50% der Übungspunkte werden zur Klausurteilnahme benötigt)
- Bestehen der Abschlussklausur oder der Nachklausur

Lehrveranstaltungen / SWS

Vorlesung 4 SWS (ca. 150 Studierende)
Übung 2 SWS
Übungsgruppen mit bis zu 30 Studierenden

Arbeitsaufwand

270 h = 90 h Präsenz und 180 h Eigenstudium

Modulnote

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

Lernziele / Kompetenzen

Breite Einführung in mathematisch fundierte Verfahren der Bildverarbeitung und des Maschinensehens.

Inhalt

1. Basics
 - 1.1 Image Types and Discretisation
 - 1.2 Degradations in Digital Images
2. Image Transformations
 - 2.1 Fourier Transform
 - 2.2 Image Pyramids
 - 2.3 Wavelet Transform
3. Colour Perception and Colour Spaces

4. Image Enhancement
 - 4.1 Point Operations
 - 4.2 Linear Filtering
 - 4.3 Wavelet Shrinkage, Median Filtering, M-Smoother
 - 4.4 Mathematical Morphology
 - 4.5 Diffusion Filtering
 - 4.6 Variational Methods
 - 4.7 Deblurring
5. Feature Extraction
 - 5.1 Edges
 - 5.2 Corners
 - 5.3 Lines and Circles
6. Texture Analysis
7. Segmentation
 - 7.1 Classical Methods
 - 7.2 Variational Methods
8. Image Sequence Analysis
 - 8.1 Local Methods
 - 8.2 Variational Methods
9. 3-D Reconstruction
 - 9.1 Camera Geometry
 - 9.2 Stereo
 - 9.3 Shape-from-Shading
10. Object Recognition
 - 10.1 Eigenspace Methods
 - 10.2 Moment Invariances

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- R. C. Gonzalez, R. E. Woods: Digital Image Processing. Addison-Wesley, Second Edition, 2002.
- K. R. Castleman: Digital Image Processing. Prentice Hall, Englewood Cliffs, 1996.
- R. Jain, R. Kasturi, B. G. Schunck: Machine Vision. McGraw-Hill, New York, 1995.
- R. Klette, K. Schlüns, A. Koschan: Computer Vision: Three-Dimensional Data from Images. Springer, Singapore, 1998.
- R. Klette, K. Schlüns, A. Koschan: Computer Vision: Three-Dimensional Data from Images. Springer, Singapore, 1998.
- E. Trucco, A. Verri: Introductory Techniques for 3-D Computer Vision. Prentice Hill, UpperSaddle River, 1998.

Modul Theoretische Stammvorlesung Computer Algebra					CS 573
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mindestens 1 in 2 Jahren	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Frank-Olaf Schreyer
Dozent/inn/en	Prof. Dr. Frank-Olaf Schreyer Prof. Dr. Wolfram Decker
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	Erfolgreiche Bearbeitung der Übungsaufgaben, Bestehen eines Zwischentests und der Abschluss- klausur
Lehrveranstaltungen / SWS	Vorlesung 4 SWS (50-100 Studierende) Übung 2 SWS, Übungsgruppen mit ca. 15 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Probleme aus der Praxis mit Computeralgebra lösen. Die Theorie hinter den Algorithmen.

Inhalt

Arithmetik und Algebraische Gleichungssysteme aus Geometrie, Ingenieur- und Naturwissenschaften.

- Ganzzahlige und Modulare Arithmetik, Primzahltest
- Polynomarithmetik und Faktorisierung
- Schnelle Fouriertransformation, modulare Algorithmen
- Resultanten, Gröbnerbasen
- Homotopiemethoden für numerisches Lösen
- reelle Lösungen, Sturmsche Ketten und andere Vorzeichen
- Regeln

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- J. van zu Gathen, J. Gerhard: Modern Computer Algebra,
- W. Decker, F.-O. Schreyer: Varieties, Gröbner Bases and Algebraic Curves,
- Sommese, C. Wampler: Numerical Solution of Polynomial Systems arising in Engineering and Science,
- H. Cohen: Computational Number Theory

Empfohlene Vorkenntnisse:

Mathematik für Informatiker 1, 2 oder Lineare Algebra 1,2 und Analysis 1.

Modul Theoretische Stammvorlesung Algorithms and Data Structures					CS 574
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mindestens 1 in 2 Jahren	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Kurt Mehlhorn
Dozent/inn/en	Prof. Dr. Kurt Mehlhorn, Prof. Dr. Raimund Seidel, Dr. Ernst Althaus, Dr. Ulrich Meyer
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	Erfolgreiche Bearbeitung der Übungsblätter, Be- stehen der benoteten Zwischen- und Endklausur.
Lehrveranstaltungen / SWS	Vorlesung 4 SWS (ca. 50 – 100 Studierende) Übung 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Die Studierenden kennen Standardalgorithmen für typische Problemstellungen aus den Be-
reichen Graphen, Geometrie, Strings und Optimierung. Zusätzlich beherrschen sie eine Reihe
abstrakter Methoden und Datenstrukturen mit deren Hilfe sie selbst effiziente Algorithmen
entwerfen und deren Laufzeit analysieren.

Inhalt

- Graphalgorithmen (Kürzeste Wege, Minimale Spannbäume, Maximale Flüsse, Matchings,
etc.)
- Geometrie (Konvexe Hülle, Delaunay Triangulierung, Voronoi Diagramme, Schnitte von
Liniensegmenten, etc.)
- Zeichenketten (Pattern matching, Suffixbäume, etc.)

- Generische Optimierungsmethoden (Tabu Search, Simulated Annealing, Generische Algorithmen, Lineare Programmierung, Branch-and-Bound, Dynamische Programmierung, Approximierung, etc.)
- Datastrukturen (Fibonacci Heaps, Radix Heaps, Hashing, Randomisierte Suchbäume, Segmentbäume, etc.)
- Analysetechniken: (Amortisierte Analyse, Average-Case Analyse, Potentialmethoden, etc.)

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- Cormen, Leiserson, Rivest and Stein, Introduction to Algorithms, Mc Graw Hill, 2001
- Aho, Hopcroft, Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974.
- Mehlhorn, Näher, LEDA, A platform for combinatorial and geometric computing, Cambridge Univ. Press, 1999.
- Tarjan, Data Structures and Network Algorithms, SIAM, 1983.
- Mehlhorn, Data Structures and Algorithms, Vol 1-3, Springer Verlag, 1984.
- Knuth, The Art of Computer Programming, Addison Wesley..

Empfohlene Vorkenntnisse:

Kenntnisse elementarer Algorithmen und Datenstrukturen, z.B. aus Programmierung 2 oder Informatik 5. Weiterhin Grundkenntnisse in C, C++ oder JAVA und die Fähigkeit, in C++ oder JAVA beschriebene Algorithmen zu verstehen.

Modul Theoretische Stammvorlesung Introduction to Computational Logic					CS 575
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mindestens 1 in 2 Jahren	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Gert Smolka
Dozent/inn/en	Prof. Dr. Gert Smolka
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	2 Klausuren (Mitte und Ende der Vorlesungszeit) Note wird aus Klausuren gemittelt und kann durch Leistungen in Übungen verbessert werden.
Lehrveranstaltungen / SWS	Vorlesung 4 SWS (ca. 50 – 100 Studierende) Übung 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

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
- programs as logical descriptions
- distinction program / specification
- verification based on invariants and weakest preconditions
- computability, proofs of non-computability

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

Programs and Computability

- regular programs, commands
- verification based on weakest preconditions
- invariants and verification conditions
- computability based on simple programs, halting problem, reductions, theorems of Rice, Gödel and Matiyasevich

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

Script for the lecture

Propositional and Predicate Logic

- Uwe Schöning,
Logik für Informatiker. 5. Auflage.
Spektrum Akademischer Verlag, 2000.
- L.T.F. Gamut,
Logic, language and meaning. Volume 1: Introduction to logic Univ. Chicago Press, 1991.
- Willard V. Quine,
Methods of Logic. 4th edition.
Harvard University Press, 1982.
- Melvin Fitting,
First-Order Logic and Automated Theorem Proving. 2nd edition.
Springer-Verlag, 1996.
- Jean H. Gallier,
Logic for Computer Science. Foundations of Automatic Theorem Proving.
Harper & Row, 1986.

Type Theory

- Peter B. Andrews,
An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof.
Kluwer Academic Publishers, 2002.
- J. Roger Hindley,
Basic Simple Type Theory.
Cambridge University Press, 1997.
- Fairouz Kamareddine, Twan Laan and Rob Nederpelt,
A Modern Perspective on Type Theory From its Origins Until Today.
Kluwer, 2004.
- John C. Mitchell,
Foundations for Programming Languages.
The MIT Press, 1996.

Hoare Logic

- Glynn Winskel,
The Formal Semantics of Programming Languages: An Introduction.
The MIT Press, 1993.
- David Harel, Dexter Kozen, Jerzy Tiuryn, *Dynamic Logic.*
The MIT Press, 2000.

Computability

- Neil D. Jones,
Computability and Complexity from a Programming Perspective.
The MIT Press, 1997.
- Harry R. Lewis and Christos H. Papadimitriou,
Elements of the Theory of Computation. 2nd edition.
Prentice Hall, 1998.
- Christos H. Papadimitriou,
Computational Complexity.
Addison Wesley, 1995.

History and Philosophy of Logic

- J.N. Crossley, et al.,
What is Mathematical Logic?
Dover Publications, 1990.
- Christos H. Papadimitriou,
Turing (A Novel about Computation).
The MIT Press, 2003.

Empfohlene Vorkenntnisse:

Programmierung 1, Einführung in die Theoretische Informatik

Modul Theoretische Stammvorlesung Geometrische Modellierung					CS 576
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mindestens 1 in 2 Jahren	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Hans-Peter Seidel
Dozent/inn/en	Prof. Dr. Hans-Peter Seidel, Prof. Dr. Philipp Slusallek
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	- Wöchentliche Aufgaben - Zwischenklausur - Endklausur
Lehrveranstaltungen / SWS	Vorlesung 4 SWS (ca. 50 - 100 Studierende) Übung 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Learning working knowledge of theoretical and practical methods for solving geometric modeling problems on a computer.

Inhalt

- Polynomial Curves
- Bezier and Rational Bezier Curves
- B-splines, NURBS
- Tensor Product Surfaces
- Shape Interrogation Methods
- Mesh Processing
- Multiresolution Modeling

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- G. Farin. *Curves and surfaces for Computer-Aided Geometric Design*, Academic Press
- J. Hoschek and D. Lasser. *Grundlagen der geometrischen Datenverarbeitung*, Teubner (original German version)
Fundamentals of computer aided geometric design, AK Peters (English translation)
- C. de Boor. *A practical Guide to Splines*, Springer
- N. Dyn. *Analysis of Convergence and Smoothness by the Formalism of Laurent Polynomials*. In: A. Iske, E. Quak, M. S. Floater. *Tutorials on multiresolution in geometric modelling: summer school lecture notes*.
- J. Warren and H. Weimer. *Subdivision methods for geometric design: a constructive approach*.
- P. Schröder, D. Zorin. *Subdivision for modelling and animation*. SIGGRAPH 2000 course notes

Modul Theoretische Stammvorlesung Komplexitätstheorie					CS 577
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mindestens 1 in 2 Jahren	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Markus Bläser
Dozent/inn/en	Prof. Dr. Markus Bläser, Prof. Dr. Raimund Seidel
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	Die Leistungen werden direkt durch die Mittse- mesterklausur und die Klausur nach Ende des Semesters bestimmt. Um zu einer Klausur zugelassen zu werden, muss man mindestens die Hälfte der bis dahin erzielbaren Übungspunkte erreicht haben. Bei Nicht-Bestehen auf Grund von Klausuren ist eine mündliche Prüfung möglich.
Lehrveranstaltungen / SWS	Vorlesung 4 SWS (ca. 50 - 100 Studierende) Übung 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

The aim of this lecture is to understand important concepts and methods of computational complexity theory.

The second part of the lecture teaches the fundamentals to understand recent topics and results in computational complexity theory.

Inhalt

Turing machines, time, space, determinism, nondeterminism, complexity classes, reduction and completeness, circuits, nonuniform complexity classes, polynomial time hierarchy, counting problems and classes, probabilistic computations, pseudorandomness and derandomisation, interactive proofs, probabilistic checkable proofs, optimisation problems and approximation

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- C.H. Papadimitriou, Computational Complexity, Addison Wesley, Reading, MA, 1993.
- M. Sipser, Introduction to the Theory of Computation, PWS Publishing, 1997.
- K.R. Reischuk, Komplexitätstheorie: Band I Grundlagen, Maschinenmodelle, Zeit- und Platzkomplexität, Nichtdeterminismus (Leitfäden der Informatik) Teubner 1999
- J. Hopcroft, R. Motwani und J. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison Wesley, Reading, MA, 2001.
- M. Garey und D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman and Company, New York, 1979.
- G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi. Complexity and Approximation. Combinatorial Optimization Problems and their Approximability Properties. Springer, 1999.

Modul Theoretische Stammvorlesung Kryptographie					CS 578
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mindestens 1 in 2 Jahren	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Michael Backes
Dozent/inn/en	Prof. Dr. Michael Backes
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	Je nach Teilnehmerzahl mündliche Prüfung oder Klausur
Lehrveranstaltungen / SWS	Vorlesung 4 SWS (ca. 150 Studierende) Übung 2 SWS Übungsgruppen mit bis zu 30 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Die wesentlichen Konzepte der Theorie der Kryptographie sowie deren Anwendung in der Praxis sind den Studierenden am Ende der Veranstaltungen bekannt. Die Studierenden sind in der Lage, bekannte Verfahren unterschiedlichen Anwendungen gemäß anzupassen sowie die Sicherheit kryptographischer Verfahren einzuschätzen.

Inhalt

- Geschichte und moderne Themen der Kryptographie
- Symmetrische und asymmetrische Verschlüsselung
- Digitale Unterschriften und Authentisierungsverfahren
- Kryptographisches Hashing und Random Oracle Model
- Informations- und komplexitätstheoretische Sicherheitsdefinitionen / Kryptographische Reduktionsbeweise

- Zentrale kryptographische Bausteine, z. B. Pseudozufallszahlengeneratoren, Trapdoor-Einwegfunktionen etc.
- Kryptographie in der Praxis (Standards, Produkte)
- Ausgewählte Themen aus der aktuellen Forschung

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- D.R. Stinson. *Cryptography – Theory and Practice*. CRC Press, 2002.
- N. Smart. *Cryptography*. McGraw-Hill Education, 2002.
- Menezes, P. van Oorschot, S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997, <http://www.cacr.math.uwaterloo.ca/hac/>
- O. Goldreich. *Foundations of Cryptography, Vol. 1: Basic Tools + Vol II: Basic Applications*. Cambridge Univ. Press, 2001 + 2004

Empfohlene Vorkenntnisse:

- Grundkenntnisse in theoretischer Informatik;
- Zahlentheoretische und komplexitätstheoretische Kenntnisse (sind nützlich aber nicht zwingend erforderlich)
- Grundvorlesungen Informatik

Modul Theoretische Stammvorlesung Optimierung					CS 579
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mindestens 1 in 2 Jahren	1 Semester	6	9

Modulverantwortliche/r	Studiendekan bzw. Studienbeauftragter der FR
Dozent/inn/en	Dr. Fritz Eisenbrand
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	Erfolgreiche Bearbeitung der Übungsaufgaben, Bestehen eines Zwischentests und der Abschluss- klausur
Lehrveranstaltungen / SWS	Vorlesung 4 SWS (ca. 50 - 100 Studierende) Übung 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Die Studierenden lernen, Optimierungsprobleme aus Theorie und Praxis zu modellieren und optimal bzw. approximativ zu lösen.

Inhalt

- Lineare Programmierung: Polyedertheorie, Simplex Algorithmus, Dualität, Ellipsoidmethode
- Ganzzahlige Programmierung: Branch-and-Bound, Schnittebenen, TDI-Systeme
- Netzwerkflußprobleme: Berechnung eines Flusses mit minimalen Kosten, Minimum Mean Cycle Augmentation, Netzwerk Simplex,
- Paarungen in Graphen: Polynomielle Matchingalgorithmen, ganzzahligkeit des Matchingpolytops, Schnittebenenverfahren
- Approximationsalgorithmen: LP-Rounding, Greedy Methoden, Knapsack, Bin Packing, Steiner Bäume, Survivable Network Design

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- Bernhard Korte, Jens Vygen: Combinatorial Optimization, Theory and Algorithms, Springer Verlag, 2001
- Alexander Schrijver: Theory of Linear and Integer Programming, Wiley-Interscience, 1986
- Alexander Schrijver: Combinatorial Optimization, Springer Verlag, 2002

Empfohlene Vorkenntnisse:

Mathematik für Informatiker, Theoretische Informatik

Modul Theoretische Stammvorlesung Semantics					CS 580
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	mindestens 1 in 2 Jahren	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Gert Smolka
Dozent/inn/en	Prof. Dr. Gert Smolka
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	2 Klausuren (Mitte und Ende der Vorlesungszeit) Note wird aus Klausuren gemittelt und kann durch Leistungen in Übungen verbessert werden.
Lehrveranstaltungen / SWS	Vorlesung 4 SWS (ca. 50 – 100 Studierende) Übung 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

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

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

Benjamin C. Pierce, Types and Programming Languages, The MIT Press, 2003.

Empfohlene Vorkenntnisse:

Introduction to Computational Logic

Modul Theoretische Stammvorlesung Verifikation					CS 581
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	7 - 10	mindestens 1 in 2 Jahren	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Holger Hermanns
Dozent/inn/en	Prof. Dr. Holger Hermanns, Prof. Dr. Bernd Finkbeiner
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	Erfolgreiche Bearbeitung der Übungsaufgaben, Bestehen einer Abschlussklausur. .
Lehrveranstaltungen / SWS	Vorlesung 4 SWS (ca. 50 - 100 Studierende) Übung 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Die Studierenden kennen die Standardansätze der computergestützten Verifikation. Sie sind mit den theoretischen Grundlagen der Methoden vertraut und sind in der Lage die Vor- und Nachteile verschiedener Methoden für ein konkretes Verifikationsproblem einzuschätzen. Die Studierenden besitzen erste praktische Erfahrungen mit manuellen Korrektheitsbeweisen und mit dem Einsatz von Verifikationswerkzeugen.

Inhalt

- Berechnungsmodelle und Spezifikationssprachen: temporale Logiken, Automaten über unendlichen Objekten, Prozeßalgebra
- Deduktive Verifikation: Beweissysteme (z.B. Floyd, Hoare, Manna/Pnueli), relative Vollständigkeit, Kompositionalität
- Model Checking: Komplexität der Model Checking Algorithmen, symbolisches Model Checking, Abstraktion
- Fallstudien

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- Doron A. Peled: Software Reliability Methods, Springer Verlag, 2001
- Zohar Manna, Amir Pnueli: Temporal Verification of Reactive Systems – Safety, Springer Verlag, 1995.
- Edmund M. Clarke, Jr., Orna Grumberg and Doron A. Peled: Model Checking, MIT Press, 2000.

Modul Theoretische Stammvorlesung Effective Computational Geometry for Curves and Surfaces					CS 650/ECG
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	7 - 10	mindestens 1 in 2 Jahren	1 Semester	4	6

Modulverantwortliche/r	Prof. Dr. Kurt Mehlhorn
Dozent/inn/en	Lutz Kettner, Kurt Mehlhorn, Susanne Schmitt, Nicola Wolpert
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung
Leistungskontrollen / Prüfungen	Homeworks, Oral Exam
Lehrveranstaltungen / SWS	Vorlesung 4 SWS
Arbeitsaufwand	180 h = 60 h Präsenz und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Our goal is the development of data structures and of efficient and exact algorithms for boolean operations on curved polygons and curved polyhedra.

Inhalt

In the lecture we address common problems in the implementation of algorithms in computational geometry, in particular, new questions when known methods for segments and lines are extended to curves and surfaces. We start with the traditional sweep-line algorithm and randomized- incremental construction. We discuss arithmetic precision, separation bounds, floating point filters, computation with algebraic numbers, curves and curve arrangements, quadric surfaces and surface arrangements, software structure of LEDA and CGAL, and C++ techniques.

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

An updated list of relevant literature will be published at the beginning of the lecture.

Recommended prerequisites:

Related core lecture Algorithms and Data Structures, Linear Algebra

Modul Theoretische Stammvorlesung Algorithms For Large Data Sets					CS 650/XXL
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	7 - 10	mindestens 1 in 2 Jahren	1 Semester	6	9

Modulverantwortliche/r

Prof. Dr. Kurt Mehlhorn

Dozent/inn/en

Dr. Ulrich Meyer

Zuordnung zum Curriculum

7. –9. Semester
Wahlpflicht / Theoretische Stammvorlesung
(LAG, LAB 1. Fach)

Zulassungsvoraussetzungen

Keine

Leistungskontrollen / Prüfungen

- Successful participation in the tutorials, the 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

Vorlesung 4 SWS
Übung 2 SWS

Arbeitsaufwand

270 h = 90 h Präsenz und 180 h Eigenstudium

Modulnote

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

Lernziele / Kompetenzen

The students know standard models, algorithms, and data structures for parallel and external-memory computing.
They know how to apply a number of abstract methods to develop and analyse efficient algorithms themselves.

Inhalt

- PRAM Algorithms (Searching and Sorting, Graphs, Geometry, Strings)
- Algorithms for Reconfigurable Networks
- External Memory Computing (Batched processing, Online processing, Paradigms for graph algorithms, Paradigms for cache-oblivious algorithms, Relationship between Parallel and External-Memory Algorithms)
- Elementary Distributed Algorithms

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- Abello, Vitter (Eds.), External Memory Algorithms, Dimacs Series Vol 50, AMS, 1999.
- Gibbons, Rytter, Efficient Parallel Algorithms, Cambridge, 1988.
- JaJa, An Introduction to Parallel Algorithms, Addison Wesley, 1992.
- Meyer, Sanders, Sibeyn (Eds.), Algorithms for Memory Hierarchies, Springer, 2003.

Recommended prerequisites:

Related core lecture Algorithms and Data Structures, Linear Algebra

Modul Theoretische Stammvorlesung Intelligent Information Agents for the Internet and Web					CS 650/I2A
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	7 - 10	mindestens 1 in 2 Jahren	1 Semester	2	4

Modulverantwortliche/r Prof. Dr. Wolfgang Wahlster, Dr. Matthias Klusch

Dozent/inn/en Dr. Matthias Klusch

Zuordnung zum Curriculum 7. –9. Semester
Wahlpflicht / Theoretische Stammvorlesung
(LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Keine

Leistungskontrollen / Prüfungen Regular attendance of classes
Passing the final oral exam

Lehrveranstaltungen / SWS Vorlesung 2 SWS

Arbeitsaufwand 120 h = 30 h Präsenz und 90 h Eigenstudium

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

Lernziele / Kompetenzen

This course presents basic and advanced concepts, methods, and systems of intelligent information agents that are useful for the development of a variety of intelligent applications in the Internet, Web, and semantic Web.

Inhalt

An information agent is a computational software entity that has access to one or multiple, heterogeneous, and distributed data and information sources; proactively searches for, mediates, and maintains relevant information on behalf of its human users or other agents, preferably just-in-time. This course presents basic and advanced concepts, methods, and systems of intelligent information agents. After a brief introduction into the domain, we survey prominent approaches to searching the Web, including methods and techniques for content-based, link-directed, and structured search, as well as various means of recommendation such as social filtering. We then turn our attention to the vision of the semantic Web, and basic means of searching by use of semantic Web agents. In addition, we also comment on the relationship between intelligent information agents and both conventional and semantic Web services. For this purpose, we start with briefly introducing the conventional Web service framework as it is exemplified by the triple SOAP-WSDL-UDDI, and surveying approaches to mediating such services in the Internet by use of middle agents such as matchmakers and

brokers. As an alternative approach, we show how semantic Web services are described in, for example, OWL-S or WSMO, and discovered by semantic Web matchmaker agents. Finally, we discuss selected methods for negotiating game-theoretically stable coalitions between information agents that charge users for any item or service they provide in order to increase monetary profits.

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

List of selected literature is provided on the course's Web page

Recommended prerequisites:

Related core lecture Artificial Intelligence

Modul Theoretische Stammvorlesung Mathematical Assistant Systems					CS 650/MAS
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	7 - 10	mindestens 1 in 2 Jahren	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Jörg Siekmann, Dr. Christoph Benz Müller
Dozent/inn/en	Dr. Serge Autexier, Dr. Christoph Benz Müller, Dr. Chad Brown, Dr. Armin Fiedler
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung
Leistungskontrollen / Prüfungen	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials • Passing the final exam • 50% of weekly exercises required to enter final exam; • 50% of final exam (written or oral) required to pass the course. • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Lehrveranstaltungen / SWS	Vorlesung 4 SWS Übung 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 90 h Präsenz und 180 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Development of a theoretical background on and some practical experience with selected mathematical assistance environments and tools

Inhalt

Part I: Introduction/Motivation:

History of Logical Reasoning Landscape of the Field Applications of Deduction Systems

Part II: Classical Deduction Systems

- Logical Foundations: - Higher-Order Logic: Syntax, Semantics - First-Order Logic: Syntax, Semantics, Calculi, Substitutions, Term-Indexing - Higher-Order Logic: Calculi I (Natural Deduction, Sequent, Intercalation, Cut-Elimination) - Higher-Order Logic: Unification - Higher-Order Logic: Calculi II (Resolution, Tableaux, Matrix, Core calculus) - Higher-Order Logic: Example Proofs (epsilon-delta-proofs, Cantor's Theorem) - Higher-Order Logic: Expansion Trees and Free Variable Semantics
- Architectures and Strategies: - Rewrite-based Theorem Proving Systems - Tactical Interactive Theorem Proving - Classical Automated First-Order and Higher-Order Theorem Proving - First-Order Logic and Induction: Explicit Induction, Rippling, Descente Infinie

Part III: Towards a Mathematical Assistant System:

What is missing for a Mathematical Assistant System? - Knowledge Based Proof Planning - Agent-based Theorem Proving - Integration of External Systems - Proof Presentation - Support for Formal Methods - Concurrent State-Based Systems - Tutorial Natural Language Dialog on Proofs - Mathematical Knowledge Representation & Management.

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

Selected papers on theory, systems and tools will be announced before the lecture starts.

Recommended prerequisites:

Some background in first-order logic and first-order reasoning. Core lecture: Introduction to Computational Logic

Modul Theoretische Stammvorlesung Natural Language Generation					CS 750/NLG
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	7 - 10	jährlich	1 Semester	4	6

Modulverantwortliche/r	Prof. Dr. Jörg Siekmann, Dr. Armin Fiedler
Dozent/inn/en	Dr. Armin Fiedler
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung
Leistungskontrollen / Prüfungen	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials • Depending on the number of students there will be an oral or written exam. • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Lehrveranstaltungen / SWS	Vorlesung 2 SWS Übung 2 SWS
Arbeitsaufwand	180 h = 60 h Präsenz und 120 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Understand the principles and techniques how NLG systems can be built.

Inhalt

Natural language generation (NLG) is a subfield of artificial intelligence and computational linguistics that is concerned with building computer software systems that can produce meaningful texts in a human language from some underlying nonlinguistic representation of information. NLG systems use knowledge about language and the application domain to automatically produce documents, reports, help messages, explanations and other kinds of texts. In this course, we will discuss theoretical issues and models of NLG. We will also discuss engineering issues that are important when building a system and examine some NLG systems.

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- E. Reiter and R. Dale, Building Natural Language Generation Systems. Cambridge University Press, Cambridge, UK, 2000
- Lecture Notes

Recommended prerequisites:

Related core lecture Artificial Intelligence basic knowledge in Artificial Intelligence and Computer Linguistics

Modul Theoretische Stammvorlesung Educational Technologies					CS 750/ET
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	7 - 10	jährlich	1 Semester	4	6

Modulverantwortliche/r	PD Dr. Erica Melis
Dozent/inn/en	PD Dr. Erica Melis
Zuordnung zum Curriculum	7. –9. Semester Wahlpflicht / Theoretische Stammvorlesung (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung
Leistungskontrollen / Prüfungen	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials • Students present the results of their Woz experiments. • Projects performed by small groups are evaluated. • A written exam may be introduced in addition.
Lehrveranstaltungen / SWS	Vorlesung 2 SWS Übung / Projekte 2 SWS Übungsgruppen mit bis zu 20 Studierenden Projektgruppen mit max. 3 Studierenden
Arbeitsaufwand	180 h = 60 h Präsenz und 120 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

After the course students should know how to design an intelligent tutoring system and adaptive hypermedia. Moreover, they should understand some of the cognitive prerequisites of building such systems.

Inhalt

- Components and architectures of educational systems
- Knowledge representation
- Student modeling
- Instructional design
- Diagnosis of student actions
- Cognitive tools

- Adaptive hypermedia, web-technologies Natural language tutorial dialogues
- Modeling tutorial strategies
- Action analysis based on Machine Learning techniques
- Computer supported collaborative learning
- Evaluation methods

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

Recent AIED and Journal articles.

Recommended prerequisites:

Related core lecture Artificial Intelligence

Modul Computer Architecture 2, Advanced Course					CS 650 / CAR2
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	Jährlich	1 Semester	6	9

Modulverantwortliche/r

Prof. Dr. W. J. Paul

Dozent/inn/en

Prof. Dr. W. J. Paul

Zuordnung zum Curriculum

Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen

Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

Studying:

Students should listen to the lectures, read the lecture notes afterwards and understand them.

They should solve the exercises alone or in groups. Students must present and explain their solutions during the tutorials.

Exams:

students who have solved 50 % of all exercises are allowed to participate in an oral exam

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

Lehrveranstaltungen / SWS

Lecture 4 h weekly, 50-100 students

Tutorials 2 h weekly, up to 20 students

Arbeitsaufwand

270 hours = 90 h classes and 180 h private study

Modulnote

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

Lernziele / Kompetenzen

After this lecture students know how to design IEEE compatible floating point units and some form of parallel computer system.

Inhalt

General comment: constructions are usually presented together with correctness proofs;
Below you find the 2005/2006 Version of this lecture

- Basics of Floating Point Computation
 - IEEE standard
 - Theory of rounding
- FPU construction
 - Add/subtract unit
 - Multiply/divide unit
 - Rounding
- Automotive systems hardware
 - Serial interfaces
 - Clock Synchronization
 - FlexRay like Interfaces
 - Electronic control units
- Automotive systems software
 - An OSEKTime like programming model
 - An OSEKTime like real time operating system
 - Drivers
 - Worst Case Execution Time
 - Pervasive Correctness proof

Weitere Informationen

Unterrichtssprache: English

Müller-Paul: Computer Architecture: Correctness and Complexity; Springer 2000

New scientific papers

Recommended prerequisites:

Related core lecture Computer Architecture

Modul Telecommunications II, Advanced Course					CS 650 / TC II
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	Jährlich	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr.-Ing. Thorsten Herfet
Dozent/inn/en	Lecture: Prof. Dr.-Ing. Thorsten Herfet Tutorial task sheets: Muhammad-Rafey Jameel, M.Sc. Tutorial: Karim Helwani (Student Assistant)
Zuordnung zum Curriculum	Wahl (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung
Leistungskontrollen / Prüfungen	Regular attendance of classes and tutorials Passing the final exam Oral exam directly succeeding the course. Eligibility: Weekly excersises / 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	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Arbeitsaufwand	270 hours = 90 h classes and 180 h private study
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

TC II will deepen the students' knowledge on modern communications systems and will focus on wireless systems.

Since from a telecommunications perspective the combination of audio/visual data – meaning inherently high data rate and putting high requirements on the realtime capabilities of the underlying network – and wireless transmission – that is unreliable and highly dynamic with respect to the channel characteristics and its capacity – is the most demanding application domain.

Inhalt

As the basic principle the course will study and introduce the building blocks of wireless communication systems. Multiple access schemes like TDMA, FDMA, CDMA and SDMA are introduced, antennas and propagation incl. link budget calculations are dealt with and more advanced channel models like MIMO are investigated. Modulation and error correction technologies presented in Telecommunications I will be expanded by e.g. turbo coding and receiver architectures like RAKE and BLAST will be introduced. A noticeable portion of the lecture will present existing and future wireless networks and their extensions for audio/visual data. Examples include 802.11 (with the TGe Quality of Service extensions), 802.16a and the terrestrial DVB system (DVB-T, DVB-H).

Weitere Informationen

Unterrichtssprache: English

Literature:

Foreground (TC II)

- Aura Ganz, Zivi Ganz, Kitty Wongthavarawat: "Multimedia Wireless Networks – Technologies, Standards, and QoS", Prentice Hall, 2004
- Simon Haykin, Michael Moher: "Modern Wireless Communications", Prentice Hall, 2005
- Ulrich Reimers: "Digital Video Broadcasting – The Family of International Standards for Digital Video Broadcasting", Springer, 2005
- William Stallings: "Wireless Communications & Networks 2nd Edition", Prentice Hall, 2005

Background (TC I)

- John G. Proakis, Masoud Salehi: "Communication Systems Engineering 2nd Edition", Prentice Hall, 2002
- Claude E. Shannon, Warren Weaver: "The Mathematical Theory of Communication", University of Illinois Press, 1963

Recommended prerequisites:

Solid foundation of mathematics (differential and integral calculus) and probability theory. The course will build on the mathematical concepts and tools taught in TC I while trying to enable everyone to follow and to fill gaps by an accelerated study of the accompanying literature.

"Signals and Systems" as well as "TC I - Digital Transmission and Signal Processing" are strongly recommended but not required.

Related core lecture TC I

Modul Database Theory, Advanced Course					CS 650 / DBT
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	Jährlich	1 Semester	4	6

Modulverantwortliche/r Prof. Dr. Christoph Koch

Dozent/inn/en Prof. Dr. Christoph Koch

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

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

Lehrveranstaltungen / SWS Lecture 3 h (weekly)
Tutorial 1 h (weekly)

Arbeitsaufwand 180 h = 60 h classes and 120 h private study

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

Lernziele / Kompetenzen

This course introduces the theory that has developed around the logical level of databases. The field of database theory draws on several areas, ranging from mathematical logic and complexity theory to programming languages. The course has two main objectives: To provide students with a solid background in the classical core material and to introduce students to recent results in the research of query languages for semistructured data.

Inhalt

- Background: logics, automata, languages, and complexity
- Theory of query languages: first-order queries, conjunctive queries, datalog, ...
 - Equivalence of Relational Algebra and Calculus (FO queries): Codd's Theorem
 - Complexity of queries
 - Query optimization: conjunctive query containment and minimization, Chase procedure
 - Recursive queries: datalog and fixpoint queries
 - (Hyper)tree decompositions, efficient processing of conjunctive queries, game-theoretic characterizations, constraint satisfaction
- Introduction to finite model theory:
 - classical vs. finite model theory: Trakhtenbrot's theorem, ...
 - Inexpressibility: Ehrenfeucht-Fraïssé-Games, pebble games, 0-1-Laws, ...
 - Descriptive complexity: Fagin's Theorem, ...
- Theory of query languages for semistructured data
 - Theory of tree automata, connections to logics
 - XML validation
 - Complexity and Expressive Power of XPath
 - XQuery: Complexity, expressive power, connections to complex-value query languages
 - Tree transducers, XSLT

Weitere Informationen

Unterrichtssprache: English

An updated list of literature will be handed out at the beginning of the lecture.

Recommended prerequisites:

Related core lecture Database Systems

Modul Automata, Games and Verification, Advanced Course					CS 650 / AG&V
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	Jährlich	1 Semester	4	6

Modulverantwortliche/r Prof. Dr. Bernd Finkbeiner

Dozent/inn/en Prof. Dr. Bernd Finkbeiner

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Englischkenntnisse gemäß § 5 des
Fachspezifischen Anhangs zur Studienordnung
und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

- Regular attendance of classes and tutorial
- Final exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester

Lehrveranstaltungen / SWS Lecture 2 h (weekly)
Tutorial 2 h (weekly)

Arbeitsaufwand 180 h = 60 h classes and 120 h private study

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

Lernziele / Kompetenzen

The students will gain a deep understanding of the automata-theoretic background of automated verification and program synthesis.

Inhalt

The theory of automata over infinite objects provides a succinct, expressive and formal framework for reasoning about reactive systems, such as communication protocols and control systems. Reactive systems are characterized by their nonterminating behaviour and persistent interaction with their environment.

In this course we study the main ingredients of this elegant theory, and its application to automatic verification (model checking) and program synthesis.

- Automata over infinite words and trees (omega-automata)
- Infinite two-person games
- Logical systems for the specification of nonterminating behavior
- Transformation of automata according to logical operations

Weitere Informationen

Unterrichtssprache: English

- Automata, Logics, and Infinite Games: A Guide to Current Research by Erich Grädel, Wolfgang Thomas, Thomas Wilke (Eds.) Lecture Notes in Computer Science 2500
- Automata Theory and its Applications by Bakhadyr Khoussainov, Anil Nerode Birkhauser Boston; 1st edition (February 15, 2001), ISBN: 0817642072

Recommended prerequisites:

Related core lecture Verification

Modul Semantics and Mechanization of Classical Higher-Order Logic Advanced Course					CS 650 / SEMHOL
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	Jährlich	1 Semester	4	9

Modulverantwortliche/r Prof. Dr. Jörg Siekmann, Dr. Christoph Benzmüller

Dozent/inn/en Dr. Christoph Benzmüller and Chad Brown

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

- 50% of weekly exercises required to enter final exam;
- 50% of final exam (written or oral) required to pass the course.
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS Lecture 4 h (weekly)
Tutorial 2 h (weekly)

Arbeitsaufwand 270 h = 90 h classes and 180 h private study

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

Lernziele / Kompetenzen

Basic overview on the latest development in the area

Inhalt

Topics of the course: untyped and simply-typed lambda calculus, Church's classical type theory, standard semantics, Henkin semantics, weaker notions of semantics, extensionality, calculi for classical type theory (natural deduction, sequent calculus, resolution, matrix), cut-elimination and (linear) cut-simulation, abstract consistency method, hintikka sets, saturation, model existence theorem, completeness of calculi via abstract consistency and model existence, strategies and heuristics, exercises with the higher-order theorem provers LEO and TPS.

Weitere Informationen

Unterrichtssprache: English

- C. Brown, Set Comprehension in Church's Type Theory , PhD Thesis, Carnegie Mellon University, 2004.
- C. Benzmüller, C. Brown, and M. Kohlhase, Higher Order Semantics and Extensionality . Journal of Symbolic Logic. (2004) 69(4):1027-1088. JSTOR
- Peter B. Andrews, An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof, Second Edition, Kluwer Academic Publishers, 2002.

Recommended prerequisites:

Some background in first-order logic and first-order reasoning. Lecture: Introduction to Computational Logic

Modul Automated Debugging, Advanced Course					CS 650 / AutoD
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	Jährlich	1 Semester	4	6

Modulverantwortliche/r Prof. Dr. Andreas Zeller

Dozent/inn/en Prof. Dr. Andreas Zeller

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

- Project exercises during the course
- Oral exam at end of course
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS Lecture 2 h (weekly)
Tutorial 2 h (weekly)

Arbeitsaufwand 180 h = 60 h classes and 120 h private study

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

Lernziele / Kompetenzen

This is a course about bugs in computer programs, how to reproduce them, how to find them, and how to fix them such that they do not occur anymore. This course teaches a number of techniques that allow you to debug any program in a systematic, and sometimes even elegant way. Moreover, the techniques can widely be automated, which allows you to let your computer do most of the debugging.

Once you understand how debugging works, you won't think about debugging in the same way. Instead of seeing a wild mess of code, you will think about causes and effects, and you will systematically set up and refine hypotheses to track failure causes. Your insights may even make you set up your own automated debugging tool. All of this allows you to spend less time on debugging, which is why you're interested in automated debugging in the first place, right?

Inhalt

Questions this course addresses include:

- How can I reproduce failures faithfully?
- How can I isolate what's relevant for the failure?
- How does the failure come to be?
- How can I fix the program in the best possible way?

Weitere Informationen

Unterrichtssprache: English

- Andreas Zeller: Why Programs Fail. Morgan Kaufmann 2005

Recommended prerequisites:

Programming skills as acquired at the Bachelor level

Modul Embedded Systems, Advanced Course					CS 650 / ES
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 10	Jährlich	1 Semester	4	9

Modulverantwortliche/r Prof. Dr. Andreas Zeller

Dozent/inn/en Prof. Dr. Andreas Zeller

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Leistungskontrollen / Prüfungen

- Written exam at the end of the course.
- Demonstration of the implemented system.
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS Lecture 4 h (weekly)
Tutorial 2 h (weekly)

Arbeitsaufwand 270 h = 90 h classes and 180 h private study

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

Lernziele / Kompetenzen

The students should learn methods for the design, the implementation, and the validation of safety-critical embedded systems.

Inhalt

Embedded Computer Systems are components of a technical system, e.g. an air plane, a car, a household machine, a production facility. They control some part of this system, often called the plant, e.g. the airbag controller in a car controls one or several airbags. Controlling means obtaining sensor values and computing values of actuator signals and sending them. Most software taught in programming courses is transformational, i.e. it is started on some input, computes the corresponding output and terminates. Embedded software is reactive, i.e. it is continuously active waiting for signals from the plant and issuing signals to the plant. Many embedded systems control safety-critical systems, i.e. malfunctioning of the system will in general cause severe damage. In addition, many have to satisfy real-time requirements, i.e. their reactions to input have to be produced within fixed deadlines.

According to recent statistics, more than 99% of all processors are embedded. Processors in the ubiquitous PC are a negligible minority. Embedded systems have a great economical impact as most innovations in domains like avionics, automotive are connected to advances in computer control. On the other hand, failures in the design of such systems may have disastrous consequences for the functioning of the overall system. Therefore, formal specification techniques and automatic synthesis of software are used more than in other domains.

The course will cover most aspects of the design and implementation of embedded systems, e.g. specification mechanisms, embedded hardware, operating systems, scheduling, validation methods.

Weitere Informationen

Unterrichtssprache: English

- Peter Marwedel: Embedded System Design, Kluwer, 2003
- Gorgio Buttazzo: Hard Real-Time Computing Systems, Springer, 2005
- Heinz Wörn, Uwe Brinkschulte: Echtzeitsysteme, Springer, 2005

Modul Data Networks II / Verification II, Advanced Course					CS 650 / DNII&VII
Studiensem. 5 - 9	Regelstudiensem. 5 - 9	Turnus zweijährlich	Dauer 1 Semester	SWS 4	ECTS-Punkte 6

Modulverantwortliche/r

Prof. Dr.-Ing. Holger Hermanns

Dozent/inn/en

Prof. Dr.-Ing. Holger Hermanns

Zuordnung zum Curriculum

Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen

Englischkenntnisse gemäß § 5 des
Fachspezifischen Anhangs zur Studienordnung
und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

Oral examination at the end of the lecture, with re-
exam possibility two month later. Permittance to
the final exam requires continuous participation in
the module, in particular:

- During the exercises, the student must successfully demonstrate at least one calculation on the blackboard.
- At the start of a part of the lectures, there are short quizzes (2–3 multiple choice questions) beingt posed. The student needs to earn at least 50% of the points for the quiz questions.

Lehrveranstaltungen / SWS

Lecture 2 h (weekly)
Tutorial 2 h (weekly)
Take home exercises 5 h per week
Self study 3 h per week

Arbeitsaufwand

180 h = 60 h classes and 120 h private study

Modulnote

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

Lernziele / Kompetenzen

The course guides through a wealth of contemporary research trends in the design of distributed systems with predictable quality of service (QoS). The students learn to understand, use, and appreciate formal methods applied to networking problems.

Inhalt

The course consists of two parts.

A first preparatory part is a crash course that brings the student to the required level for the main part if only one of the required courses has been followed.

For those who did not follow "Data Networks", the course reviews:

- basic probability theory,
- stochastic models, Markov chains,
- protocol design fundamentals,
- reliable data transfer.

For those who did not follow "Verification", the course reviews:

- principles of model checking,
- transition systems and Kripke structures,
- temporal logics,
- computation tree logic.

The main part of the module then covers:

Formal models

- process algebra,
- timed automata
- generalized semi-Markov processes,
- stochastic process algebra,
- Modest,
- UML statecharts and extensions.

Model checking

- real-time model checking,
- Markov model checking,
- simulation-based model checking.

Applications

- GSM-R train radio,
- lacquer production planning,
- universal plug-and-play protocols,
- extasy optimisation (if time permits).

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

James F. Kurose, Keith W. Ross: Computer networking : a top-down approach featuring the internet. – 2nd ed. – Boston : Addison-Wesley, 2003

- Dimitri Bertsekas, Robert Gallager: Data networks. – 2nd ed. – Saddle River, NJ : Prentice Hall, 1992.
- Gerald J. Holzmann: The spin model checker : primer and reference manual. – Boston : Addison-Wesley, 2004.
- Gerald J. Holzmann: Design and Validation of Computer Protocols. – Englewood Cliffs, NJ : Prentice–Hall, 1991.
- Andrew S. Tanenbaum: Computer networks. – 4th ed. – Upper Saddle River, NJ : Pearson Education, 2003.
- Christel Baier, Boudewijn Haverkort, Holger Hermanns, Joost-Pieter Katoen: Model-checking algorithms for continuous-time Markov chains. IEEE transactions on software engineering 29(6), 524-541. 2003.
csdl.computer.org/comp/trans/ts/2003/06/e0524abs.htm
- Holger Hermanns. Interactive Markov chains and the quest for quantified quality / Berlin : Springer, 2002. Chapters 2 to 4.
<http://link.springer.de/link/service/series/0558/tocs/t2428.htm>
- Kim G. Larsen, Paul Petterson, Wang Yi: Uppaal in a nutshell. Journal for Software Tools for Technology Transfer 1. 1997. www.docs.uu.se/docs/rtmv/papers/lpw-sttt97.ps.gz
- Rik Eshuis, Roel Wieringa: Requirements-level semantics for UML statecharts. CTIT technical report, 2000. www.ub.utwente.nl/webdocs/ctit/1/00000026.pdf
- David N. Jansen, Holger Hermanns, Joost-Pieter Katoen: A QoS-oriented extension of UML statecharts. In: <UML> 2003: The unified modeling language: modeling languages and applications. Berlin: Springer, 2003. (LNCS, 2863). pages 76-91.
<http://www.springerlink.com/link.asp?id=l1gny2dlfg8gjyx5>
- Henrik Bohnenkamp, Pedro R. D'Argenio, Holger Hermanns, Joost-Pieter Katoen: MoDeST: A compositional modelling formalism for hard and softly timed systems. CTIT TechRep. 04-46.

Recommended prerequisites:

Related core lecture “Data networking“ OR “Verification“

Modul Operating Systems Practice, Advanced Course					CS 750 / OSP
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 9	jährlich	1 Semester	2	4

Modulverantwortliche/r Prof. Dr. Helge Scheidig

Dozent/inn/en Prof. Dr. Helge Scheidig

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Englischkenntnisse gemäß § 5 des
Fachspezifischen Anhangs zur Studienordnung
und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

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

Lehrveranstaltungen / SWS Lecture 2 h (weekly)

Arbeitsaufwand 120 h = 30 h classes and 90 h private study

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

Lernziele / Kompetenzen

Operating system construction techniques
Basic skills in creating driver components of operating system kernels

Inhalt

- fundamental operating system construction techniques (monolithic kernel, micro-kernel, nucleus)
- driver architectures and their integration into monolithic operating systems (Focus on Linux-kernel drivers)
- programming operating systems kernels and programming drivers

Weitere Informationen

Unterrichtssprache: Deutsch

Literatur:

- H. Herold: "Linux/Unix Systemprogrammierung", Addison-Wesley, 2004
- R. Love: "Linux Kernel Development", Pearson, 2005
- J. Quade, E.-K. Kunst: "Linux-Treiber entwickeln", dpunkt.Verlag, 2004

Recommended prerequisites:

Related core lecture Operating Systems

Modul Distributed Operating Software, Advanced Course					CS 750 / DOS
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 9	jährlich	1 Semester	2	4

Modulverantwortliche/r Prof. Dr. Helge Scheidig

Dozent/inn/en Prof. Dr. Helge Scheidig

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

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

Lehrveranstaltungen / SWS Lecture 2 h (weekly)

Arbeitsaufwand 120 h = 30 h classes and 90 h private study

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

Lernziele / Kompetenzen

Concepts and techniques in distributed operating software

Inhalt

Central characteristics, differences regarding traditional operating systems

- classification, types of architecture
- basic components and services (lightweight processes and team-concept, communication systems, process-synchronisation, equipment administration, bug tolerance, failure reliability, naming, filing)

Remote procedure call

Coordination and coordination platforms

Weitere Informationen

Unterrichtssprache: Deutsch

Literatur:

An updated list of literature will be handed out at the beginning of the lecture.

Recommended prerequisites:

Related core lecture Operating Systems

Modul Computer Graphics II, Advanced Course Realistic Image Synthesis					CS 650 / CGII-RIS
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 9	zweijährlich	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Philipp Slusallek

Dozent/inn/en Prof. Dr. Philipp Slusallek

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

- Theoretical and practical exercises (50% requirement for final exam)
- Final oral exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS Lecture 4 h (weekly)
Tutorial 2 h (weekly)

Arbeitsaufwand 270 h = 90 h classes and 180 h private study

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

Lernziele / Kompetenzen

At the core of computer graphics is the requirement to render highly realistic and often even physically accurate images of virtual 3D scenes. In this lecture students will learn about physically-based simulation techniques to compute the distribution of light in even complex environment. After this course students should be able to build their own highly realistic but also efficient rendering system.

Inhalt

- Rendering and Radiosity Equation, Finite Elements
- Radiosity
- Monte Carlo Techniques
- Direct Illumination, Importance Sampling
- BRDF, Inversion Methods
- Distribution Ray Tracing and Path Tracing
- Theory of Variance Reduction
- Bidirectional Path Tracing, Instant Radiosity
- Density Estimation Methods
- Photon Mapping
- Rendering of Animations
- Motion Blur, Temporal Filtering
- Interactive Global Illumination
- Hardware Rendering Basics
- Advanced Hardware Rendering
- Measurements of BRDFs and Light Sources
- Relighting
- Tone Mapping, Perception

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- Michael Cohen, John Wallace, Radiosity and Realistic Image Synthesis, Academic Press, 1993.
- Andrew Glassner, Principles of Digital Image Synthesis, 2 Bände, Morgan Kaufman, 1996.
- Andrew Glassner, An Introduction to Ray Tracing, Academic Press, 1989.
- James Foley, Andries Van Dam, et al., Computer Graphics: Principles and Practice, 2. Ausgabe, Addison-Wesley, 1995

Recommended prerequisites:

Related core lecture Computer Graphics

Modul Computer Graphics II, Advanced Course 3D Image Analysis and Synthesis					CS 650 / CG II-3D
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 9	zweijährlich	1 Semester	4	6

Modulverantwortliche/r Prof. Dr. Hans-Peter Seidel, Dr. Marcus Magnor

Dozent/inn/en Dr. Marcus Magnor, Dr. Volker Blanz

Zuordnung zum Curriculum Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

- Succesfull completion of all assignments
- Oral exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester

Lehrveranstaltungen / SWS Lecture 2 h (weekly)
Tutorial 2 h (weekly)

Arbeitsaufwand 180 h = 60 h classes and 120 h private study

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

Lernziele / Kompetenzen

- Imaging fundamentals
- Reconstruction algorithms
- Implementation in C/C++
- Rendering from image data

Inhalt

- Image formation
- Photometric and geometric camera calibration
- Low-level image processing
- 3D scanning
- reflection properties
- 3d reconstruction, image-based rendering, motion capture

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- Reinhard Klette, Andreas Koshan, Karsten Schins, „Computer Vision“, Vieweg 1996
- Richard Hartley and Andrew Zisserman, „Multiple View Geometry in Computer Vision“

Recommended prerequisites:

Related core lecture Computer Graphics

Modul Multimedia, Advanced Course					CS 650 / MM
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 9	zweijährlich	1 Semester	4	6

Modulverantwortliche/r	Prof. Dr. Philipp Slusallek
Dozent/inn/en	Prof. Dr. Philipp Slusallek, Marco Lohse
Zuordnung zum Curriculum	Wahl (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung.
Leistungskontrollen / Prüfungen	<ul style="list-style-type: none"> • Theoretical and practical exercises (50% requirement for final exam) • A final oral exam • The final grade depend equally on the excercises 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	Lecture 2 h (weekly) Tutorial 2 h (weekly)
Arbeitsaufwand	180 h = 60 h classes and 120 h private study
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Digital media has become an integral part of today's desktop computers and mobile systems allowing us to capture, create, process, edit, render audio and video in realtime. This lecture will present the theoretical, technical, and practical background of handling media streams in a computer.

Of particular interest is middleware that allows for creating distributed digital media applications that can transparently use any devices or processing module available in a network. The course is based on the Network-Multimedia-System (NMM) developed in Saarbrücken.

After the course students should understand the theoretical, technological, and practical issues and challenges of digital media systems. They should be able to create their own media processing modules for NMM or similar systems.

Inhalt

- Digital media devices and their characteristics
- Perception issues
- Types and formats of multimedia streams
- Signal processing
- Audio and video compression techniques
- Local and distributed synchronization
- Network protocols and technology for streaming media
- Middleware for digital media
- Applications

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- Steinmetz, Ralf: *Multimedia-Technologie. Grundlagen, Komponenten und Systeme*; Dritte, überarbeitete Auflage, Springer, Berlin; Heidelberg; New York (2000).
- Foley, James D. and van Dam, Andries and Feiner, Steven K. and Hughes, John F.: *Computer Graphics: Principles and Practice*, second edition in C. The Systems Programming Series. Addison-Wesley, Bonn;Amsterdam;Tokyo (1996).

Modul Differential Equations in Image Processing and Computer Vision Advanced Course					CS 650 / DIC
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 9	zweijährlich	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Joachim Weickert
Dozent/inn/en	Prof. Dr. Joachim Weickert, Dr. Martin Welk, Dr. Bernhard Burgeth
Zuordnung zum Curriculum	Wahl (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung
Leistungskontrollen / Prüfungen	<ul style="list-style-type: none"> • Regular attendance of lecture and tutorial • 50% of all possible points from weekly assignments to be eligible for the final exam are needed • Passing the final exam or the re-exam • The re-exam takes place during the last two weeks before the start of lectures in the following semester
Lehrveranstaltungen / SWS	Lecture 4 h (weekly) Tutorial 2 h (weekly) 50% theoretical exercises and 50% practical programming assignments
Arbeitsaufwand	270 h = 90 h of classes and 180 h private study
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Many modern techniques in image processing and computer vision make use of methods based on partial differential equations (PDEs) and variational calculus. Moreover, many classical methods may be reinterpreted as approximations of PDE-based techniques. In this course the students will get an in-depth insight into these methods. For each of these techniques, they will learn the basic ideas as well as theoretical and algorithmic aspects. Examples from the fields of medical imaging and computer aided quality control will illustrate the various application possibilities.

Inhalt

1. Introduction and Overview
2. Linear Diffusion Filtering
 - 2.1 Basic Concepts
 - 2.2 Numerics
 - 2.3 Limitations and Alternatives
3. Nonlinear Isotropic Diffusion Filtering
 - 3.1 Modeling
 - 3.2 Continuous Theory
 - 3.2 Semidiscrete Theory
 - 3.3 Discrete Theory
 - 3.4 Efficient Sequential and Parallel Algorithms
4. Nonlinear Anisotropic Diffusion Filtering
 - 4.1 Modeling
 - 4.2 Continuous Theory
 - 4.3 Discrete Aspects
5. Parameter Selection
6. Variational Methods
 - 6.1 Basic Ideas
 - 6.2 Discrete Aspects
 - 6.3 TV Denoising, Equivalence Results
 - 6.4 Mumford-Shah Segmentation and Diffusion-Reaction Filters
7. Vector- and Matrix-Valued Images
8. Image Sequence Analysis
 - 8.1 Global Methods
 - 8.2 Local Methods
 - 8.3 Combined Local-Global Methods
 - 8.4 Numerical Techniques
9. Continuous-Scale Morphology
 - 9.1 Basic Ideas
 - 9.2 Applications
10. Curvature-Based Morphology
 - 10.1 Basic Ideas
 - 10.2 Applications

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- J. Weickert: Anisotropic Diffusion in Image Processing. Teubner, Stuttgart, 1998.
- G. Sapiro: Geometric Partial Differential Equations in Image Analysis. Cambridge University Press, 2001.
- G. Aubert and P. Kornprobst: Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations. Springer, New York, 2002.
- Articles from journals and conferences.

Recommended prerequisites:

Related core lecture Computer Vision

Modul Differential Geometric Aspects of Image Processing Advanced Course					CS 650 / DGAIP
Studiensem. 5 - 9	Regelstudiensem. 5 - 9	Turnus zweijährlich	Dauer 1 Semester	SWS 3	ECTS-Punkte 5

Modulverantwortliche/r

Prof. Dr. Joachim Weickert

Dozent/inn/en

Dr. Martin Welk

Zuordnung zum Curriculum

Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen

Englischkenntnisse gemäß § 5 des
Fachspezifischen Anhangs zur Studienordnung
und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

- Written or oral exam at end of course
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS

Lecture 2 h (weekly)
Tutorial 1 h (weekly)

Arbeitsaufwand

150 h = 45 h classes and 105 h private study

Modulnote

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

Lernziele / Kompetenzen

Specialised course in mathematical image analysis.
Participants learn how concepts of differential geometry can be applied in image processing.

Mathematical prerequisites which exceed the basic mathematics courses for computer science students are provided within the lecture.

Inhalt

The course is concerned with modern methods of digital image processing which rely on the differential geometry of curves and surfaces. This includes methods of image enhancement (like smoothing procedures) as well as feature extraction and segmentation (like locating contours using active contour models).

The course aims at combining theoretical foundation directly with a variety of applications from the above-mentioned fields; the range of topics extends up to recent research problems.

An introduction to the relevant concepts and results from differential geometry will be included in the course.

Topics include:

- curves and surfaces in Euclidean space
- level sets
- curve and surface evolutions
- variational formulations and gradient descents
- diffusion of scalar and non-scalar data
- diffusion on manifolds
- active contours and active regions.

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- F. Cao, Geometric Curve Evolution and Image Processing. Lecture Notes in Mathematics, vol. 1805, Springer, Berlin 2003.
- R. Kimmel, Numerical Geometry of Images. Springer, Berlin 2004.
- S. Osher, N. Paragios, eds., Geometric Level Set Methods in Imaging, Vision and Graphics. Springer, Berlin 2003.
- G. Sapiro, Geometric Partial Differential Equations and Image Analysis. Cambridge University Press 2001.
- Articles from journals and conferences.

Recommended prerequisites:

Related core lecture Computer Vision

Completed Mathematics for Computer Scientists lectures.

Modul Introduction to Image Acquisition Methods, Advanced Course					CS 750 / IIAM
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 9	jährlich	1 Semester	2	4

Modulverantwortliche/r	Prof. Dr. Joachim Weickert
Dozent/inn/en	Dr. Martin Welk
Zuordnung zum Curriculum	Wahl (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung
Leistungskontrollen / Prüfungen	<ul style="list-style-type: none"> • Written or oral exam at end of course • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Lehrveranstaltungen / SWS	Lecture 2 h (weekly)
Arbeitsaufwand	120 h = 30 h classes and 90 h private study
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

The course is designed as a supplement for image processing lectures, to be attended before, after or parallel to them.

Participants shall understand

- what are digital images
- how they are acquired
- what they encode and what they mean
- which limitations are introduced by the image acquisition.

This knowledge will be helpful in selecting adequate methods for processing image data arising from different methods.

Inhalt

A broad variety of image acquisition methods is described, including imaging by virtually all sorts of electromagnetic waves, acoustic imaging, magnetic resonance imaging and more. While medical imaging methods play an important role, the overview is not limited to them.

Starting from physical foundations, description of each image acquisition method extends via aspects of technical realisation to mathematical modelling and representation of the data.

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- B. Jähne, H. Haußecker, P. Geißler, editors, Handbook of Computer Vision and Its Applications. Volume 1: Sensors and Imaging. Academic Press, San Diego 1999.
- S. Webb, The Physics of Medical Imaging. Institute of Physics Publishing, Bristol 1988.
- C. L. Epstein, Introduction to the Mathematics of Medical Imaging. Pearson, Upper Saddle River 2003.
- C. Kak, M. Slaney, Principles of Computerized Tomographic Imaging. SIAM, Philadelphia 2001.
- Articles from journals and conferences.

Recommended prerequisites:

Related core lecture Computer Vision

Modul Probabilistic Methods in Image Processing, Advanced Course					CS 750 / PMIP
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 9	jährlich	1 Semester	2	4

Modulverantwortliche/r	Prof. Dr. Joachim Weickert
Dozent/inn/en	Dr. Bernhard Burgeth
Zuordnung zum Curriculum	Wahl (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung
Leistungskontrollen / Prüfungen	<ul style="list-style-type: none"> • Oral exam at the end of the semester • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Lehrveranstaltungen / SWS	Lecture 2 h (weekly)
Arbeitsaufwand	120 h = 30 h classes and 90 h private study
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.
Lernziele / Kompetenzen	The students will learn some basic knowledge from probability theory and statistics, and they will learn how to apply these concepts to image processing problems.

Inhalt

1. Probabilistic Background:
 - 1.1 Probability Measures
 - 1.2 One-dimensional Distributions
 - 1.3 Multivariate Distributions
 - 1.4 Conditional Probability
 - 1.5 Independence
 - 1.6 Random Variables
 - 1.7 Moments
 - 1.8 Characteristic Functions
 - 1.9 Limit Theorems

2. Probabilistic Concepts in Image Processing
 - 2.1 Histogram Based Techniques
 - 2.2 Co-Histograms
 - 2.3 Locally Orderless Images
 - 2.4 Registration, Entropy, and Mutual Information
 - 2.5 Parzen Estimates and Maximum Likelihood
 - 2.6 PCA and SVD

3. Optional:
 - Introduction to Markov Random Fields

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- Krengel, U.: Einführung in die Wahrscheinlichkeitstheorie (7. Aufl.). Braunschweig, Vieweg 2003.
- Jacod, J.; Protter, P.: Probability Essentials. Springer, 2000
- Behrends, E.: Introduction to Markov Chains. Braunschweig, Vieweg 2000.

Recommended prerequisites:

Related core lecture Computer Vision

Modul Pattern Recognition, Advanced Course					CS 750 / PR
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 9	jährlich	1 Semester	3	5

Modulverantwortliche/r

Prof. Dr. Joachim Weickert

Dozent/inn/en

Dr. Bernhard Burgeth

Zuordnung zum Curriculum

Wahl (LAG, LAB 1. Fach)

Zulassungsvoraussetzungen

Englischkenntnisse gemäß § 5 des
Fachspezifischen Anhangs zur Studienordnung
und zur Prüfungsordnung

Leistungskontrollen / Prüfungen

- Regular attendance of lecture and tutorial
- 50% of all possible points from weekly assignments to be eligible for the final exam are needed
- Passing the final exam or the re-exam
- The re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS

Lecture 2 h (weekly)
Tutorial 1 h (weekly)

Arbeitsaufwand

150 h = 45 h classes and 105 h private study

Modulnote

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

Lernziele / Kompetenzen

Introduction to the main concepts of kernel-based methods for pattern recognition. The students will learn the mathematical and theoretical foundations as well as suitable algorithms for analysing these data.

Inhalt

1. Basic Concepts:
 - 1.1 Pattern Analysis
 - 1.2 Kernel Methods, Overview
 - 1.3 First Properties of Kernels

2. Algorithmic Aspects:
 - 2.1 Algorithms in Feature Space
 - 2.2 Algorithms Based on Eigen-Decomposition and Optimisation

3. Construction of Kernels:
 - 3.1 Kernel Types
 - 3.2 Kernels for Strings, Trees

4. Optional:
 - Markov Models

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

- Shawe-Taylor and Christiani: Kernel Methods for Pattern Recognition. Cambridge University Press, 2004.
- Duda and Hart: Pattern Classification (2nd Ed.). New York, Wiley, 2001.
- Wissenschaftliche Artikel

Recommended prerequisites:

Related core lecture Computer Vision

Modul Seminar Changing Topics					CS 500
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5 - 9	5 - 9	jährlich	1 Semester	3	7

Modulverantwortliche/r	Dean of studies and relevant Professor
Dozent/inn/en	Professors of the Department
Zuordnung zum Curriculum	Wahl (LAG, LAB 1. Fach)
Zulassungsvoraussetzungen	Englischkenntnisse gemäß § 5 des Fachspezifischen Anhangs zur Studienordnung und zur Prüfungsordnung
Leistungskontrollen / Prüfungen	<ul style="list-style-type: none"> • Contributions to discussions • Thematic talk • Written elaboration • Final oral examination on the entire scientific area spanned by the seminar
Lehrveranstaltungen / SWS	Seminar 3 h (weekly) / groups of up to 20 students
Arbeitsaufwand	210 h = 45 h classes und 165 h private study
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

At the end of the course students have gained a thorough knowledge of current or foundational aspects of a specific area in computer science.

They attained competences in independently investigating, classifying, summarizing, discussing, criticizing scientific issues and presenting scientific findings.

Inhalt

Practical exercising of

- Reflecting on scientific work,
- Analyzing and assessing scientific papers
- Composing scientific abstracts
- Discussing scientific work in a peer group
- Developing common standards for scientific work
- Presentation techniques

Specific focus according to the individual topic of the seminar.

Typical course progression:

- Preparatory meetings to guide selection of individual topics
 - Repetitive meetings with discussions of selected contributions
 - Talk and elaboration on one of the contributions
- Oral exam on entire scientific area spanned by the seminar

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:

According to the topic

Modul Didaktik der Informatik					Ddl
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
3-7	3 - 8	i.d. R. jährlich	2 - 3 Semester	6	9

Modulverantwortliche/r	StD Peter Miroid Lehrbeauftragter für die Didaktik der Informatik
Dozent/inn/en	StD Peter Miroid Lehrbeauftragte/r für die Didaktik der Informatik
Zuordnung zum Curriculum	Pflicht
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	<ul style="list-style-type: none"> • Beiträge zur Diskussion • Thematische Vorträge • Schriftliche Ausarbeitung • Schriftliche Abschlussprüfung oder mündliches Abschlussgespräch • Praktikumsbericht (Laborpraktikum)
Lehrveranstaltungen / SWS	<ul style="list-style-type: none"> • 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) <u>oder</u> Fachdidaktisches Laborpraktikum Laborpraktikum zur Programmierung, Softwaretechnik oder zu Themen der Technischen Informatik (Robotik, Rechnerbau) (2 SWS, 3 KPe)
Arbeitsaufwand	<p>Vorlesung: 30 h Präsenz 60 h Selbststudium</p> <p>Seminare/Praktikum 60 h Präsenz 20 h Vor- und Nacharbeit</p>
Modulnote	Die Einzelnoten der Modulteile 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 Modulteile.

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

Weitere Informationen

Literatur:

Rüdeger Baumann: Didaktik der Informatik
Peter Hubwieser: Didaktik der Informatik
S. Schubert, A. Schwill: Didaktik der Informatik

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 .

Modul Elementare fachdidaktische schulpraktische Studien					ESP
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
3 – 5	3 - 7	Jährlich i.d.R im SS	1 Semester	2 SWS+ 15 Tage	7

Modulverantwortliche/r

StD Peter Miroid
Lehrbeauftragter für die Didaktik der Informatik

Dozent/inn/en

StD Peter Miroid
Lehrbeauftragter für die Didaktik der Informatik

Zuordnung zum Curriculum

Pflicht

Zulassungsvoraussetzungen

Keine

Leistungskontrollen / Prüfungen

- Erfolgreiche Teilnahme am Begleitseminar
- Praktikumsbericht

Lehrveranstaltungen / SWS

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

Arbeitsaufwand

270 Stunden:
davon 180 Stunden Schulpraktikum
(Präsenzzeit, Zeit für die Vor- und Nacharbeitung
der Unterrichtshospitationen und eigener
Unterrichtsversuche),
90 Stunden Begleitseminar
(30 h Präsenzzeit, 60 h für die Vor- und
Nacharbeitung der Seminarsitzungen sowie das
Erstellen des Praktikumsberichtes)

Modulnote

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

Weitere Informationen

Literatur:

Rüdeger Baumann: Didaktik der Informatik
Peter Hubwieser: Didaktik der Informatik
S. Schubert, A. Schwill: Didaktik der Informatik

Empfehlung:

Eine erfolgreiche Absolvierung der Pflichtmodule Programmierung 1 und 2, Systemarchitektur und Grundzüge der Theoretischen Informatik vor Besuch der Veranstaltungen ist ratsam.

Modul Vertiefte fachdidaktische schulpraktische Studien					VSP
Studiensem.	Regelstudiensem	Turnus	Dauer	SWS	ECTS-Punkte
3 – 5	3 - 7	Jährlich	1 Semester	2 SWS+ 4 Wochen	9

Modulverantwortliche/r

StD Peter Mirol
Lehrbeauftragter für die Didaktik der Informatik

Dozent/inn/en

StD Peter Mirol
Lehrbeauftragter für die Didaktik der Informatik

Zuordnung zum Curriculum

Pflicht

Zulassungsvoraussetzungen

Keine

Leistungskontrollen / Prüfungen

- Erfolgreiche Teilnahme am Begleitseminar
- Praktikumsbericht

Lehrveranstaltungen / SWS

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

Arbeitsaufwand

270 Stunden.
Davon 180 Stunden Schulpraktikum
(Präsenzzeit, Zeit für die Vor- und Nacharbeitung der Unterrichtshospitationen und eigener Unterrichtsversuche)
90 Stunden Begleitseminar
(30 h Präsenzzeit, 60 h für die Vor- und Nacharbeitung der Seminarsitzungen sowie das Erstellen des Praktikumsberichtes)

Modulnote

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 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

Weitere Informationen

Literatur:

Literatur:

Rüdeger Baumann: Didaktik der Informatik
Peter Hubwieser: Didaktik der Informatik
S. Schubert, A. Schwill: Didaktik der Informatik

Empfehlung:

Die erfolgreiche Absolvierung der Pflichtmodule des Grundstudiums und des Moduls Elementare fachdidaktische schulpraktische Studien vor Absolvierung des Moduls wird angeraten.