Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

UNIVERSITÄT DES SAARLANDES

Modulhandbuch

für den Bachelor Studiengang Informatik

(01.10.2016)

Studiensem.	Modul	CP	SWS
Ringvorlesungen über Themen der Informatik (Pflicht)			
1	Perspektiven der Informatik	2	2
Mathematische Grundlagen (Pflicht)			
1	Mathematik für Informatiker 1	9	6
2	Mathematik für Informatiker 2	9	6
3	Mathematik für Informatiker 3	9	6
Grundlagen der Informatik (Pflicht)			
1	Programmierung 1	9	6
2	Programmierung 2	9	6
2	Systemarchitektur	9	6
3	Grundzüge von Algorithmen und Datenstrukturen	6	4
4	Informationssysteme	6	4
4	Nebenläufige Programmierung	6	4
3	Grundzüge der Theoretischen Informatik	9	6
Praktikum (Pflicht)			
3	Softwaredesignpraktikum	9	6
Stammvorlesungen der Informatik (Wahlpflicht)			
5	Operating Systems	9	6
5	Computer Graphics	9	6
5	Database Systems	9	6
5	Embedded Systems	9	6
5	Information Retrieval and Data Mining	9	6
5	Artificial Intelligence	9	6

Studiensem.	Modul	CP	SWS
5	Computer Architecture	9	6
5	Security	9	6
5	Software Engineering	9	6
5	Compiler Construction	9	6
5	Automated Reasoning	9	6
5	Image Processing and Computer Vision	9	6
5	Computer Algebra	9	6
5	Algorithms and Data Structures	9	6
5	Introduction to Computational Logic	9	6
5	Geometric Modelling	9	6
5	Complexity Theory	9	6
5	Cryptography	9	6
5	Optimization	9	6
5	Semantics	9	6
5	Verification	9	6
5	Telecommunications I	9	6
5	Machine Learning	9	6
5	Distributed Systems	9	6
5	Data Networks	9	6
Vertiefungsvorlesungen der Informatik (Wahl)			
6	Computer Architecture 2	9	6
6	Telecommunications II	9	6
6	Automata, Games and Verification	6	4

Studiensem.	Modul	CP	SWS
6	Automated Debugging	6	4
6	Computer Graphics II	9	6
6	Differential Equations in Image Processing and Computer Vision	9	6
6	Introduction to Image Acquisition Methods	4	2
6	Correspondence Problems in Computer Vision	6	4
1,3	Future Media Internet	9	6
6	Automatic Planning	9	4
4	Proseminar	5	2
5	Seminar	7	3
6	Bachelor-Seminar	9	5
6	Bachelorarbeit	12	
Beliebig wählbare Module des BA-Studienganges Informatik (Wahl)			
Jedes Semester	Praktikum zum Informationsmanagement	6/9	4/6
ab 2.	Tutortätigkeit	4	2
Soft Skills			
3.	Methodik und Didaktik für Tutoren	1	30h
Sprachkurse – max. 6 CP; lebende Sprachen (Wahl)			
1.-5.	Sprachkurse	3 o. 6	2 o. 4

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Ringvorlesung Perspektiven der Informatik					**CS 101**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1	6	Jährlich / WS	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	Bachelor Informatik, Pflicht
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	Positive Bewertung von mindestens drei schriftlichen Zusammenfassungen verschiedener Vorträge
Lehrveranstaltungen / SWS	Vorlesung / 2 SWS
Arbeitsaufwand	120 h = 30 h Präsenz- und 90 h Eigenstudium
Modulnote	Das Modul ist insgesamt bestanden, wenn die Prüfungsleistung bestanden wurde (unbenotet).

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

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

Modulverantwortliche/r	Prof. Dr. Joachim Weickert
Dozent/inn/en	Prof. Dr. Joachim Weickert, Prof. Dr. Frank-Olaf Schreyer
Zuordnung zum Curriculum	Bachelor Informatik mit anderem Nebenfach, Pflicht
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
Lehrveranstaltungen / SWS	Vorlesung: 4 SWS Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 80 h Präsenz- und 190 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

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. Binomialköffizienten 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

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Mathematik für Informatiker 2					**CS 210 / Mfi2**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
2	6	Jährlich / SS	1 Semester	6	9

Modulverantwortliche/r	Prof. Prof. Dr. Joachim Weickert
Dozent/inn/en	Prof. Dr. Joachim Weickert, Prof. Dr. Frank-Olaf Schreyer
Zuordnung zum Curriculum	Bachelor Informatik mit anderem Nebenfach, Pflicht
Zulassungsvoraussetzungen	MfI 1 (empfohlen)
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
Lehrveranstaltungen / SWS	Vorlesung: 4 SWS Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 80 h Präsenz- und 190 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

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

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, Austauschsatz
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

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Mathematik für Informatiker 3					**CS 310 / Mfi3**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
3	6	Jährlich / WS	1 Semester	6	9

Modulverantwortliche/r	Prof. Prof. Dr. Joachim Weickert
Dozent/inn/en	Prof. Dr. Joachim Weickert, Prof. Dr. Frank-Olaf Schreyer
Zuordnung zum Curriculum	Bachelor Informatik mit anderem Nebenfach, Pflicht
Zulassungsvoraussetzungen	MfI1 und MfI 2 (empfohlen)
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
Lehrveranstaltungen / SWS	Vorlesung: 4 SWS Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 80 h Präsenz- und 190 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

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.

STOCHASTIK, NUMERIK UND MEHRDIMENSIONALE ANALYSIS

E. NUMERISCHE ERGÄNZUNGEN (3)

52. Banachscher Fixpunktsatz (1)
53. Interpolation, incl. Splines (2)

F. MEHRDIMENSIONALE ANALYSIS UND NUMERIK (11)

54. Stetigkeit und Differentialoperatoren für skalarwertige Funktionen (2)
55. Differentialoperatoren für vektorwertige Funktionen (1)
56. Totale Differenzierbarkeit (1/2)
57. Mittelwertsatz und Satz von Taylor (1 1/2)
58. Extrema von Funktionen mehrerer Variabler (1)
59. Das Newton-Verfahren (1)
60. Extrema mit Nebenbedingungen (1)
61. Mehrfachintegrale (1)
62. Die Umkehrfunktion und die Transformationsregel (1)
63. Variationsrechnung (1)

G. STOCHASTIK (16)

64. Grundbegriffe (Ws., Stichprobenraum) (1/3)
65. Kombinatorik (2/3)
66. Erzeugende Funktionen (1)
67. Bedingte Wahrscheinlichkeiten (1)
68. Zufallsvariable, Erwartungswert, Varianz (2) (Systemzuverlässigkeit, Varianz, Kovarianz, Jensen)
69. Abschätzungen für Abweichungen vom Mittelwert (1) (Momente, Schranken von Markov, Chebyshev, Chernoff, schwaches Gesetz der grossen Zahlen)
70. Wichtige diskrete Verteilungen (1)
71. Wichtige kontinuierliche Verteilungen (1) (incl. zentraler Grenzwertsatz)
72. Multivariate Verteilungen und Summen von Zufallsvariablen (1)
73. Parameterschätzung und Konfidenzintervalle (1)
74. Hypothesentests (1)
75. Methode der kleinsten Quadrate (1)
76. Robuste Statistik (2/3)
77. Fehlerfortpflanzung (1/3)
78. Markowketten (2)
79. Pseudozufallszahlen und Monte-Carlo-Simulation (1)

Weitere Informationen

Unterrichtssprache: deutsch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Programmierung 1					**CS 120 / P1**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1	6	Jährlich / WS	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Gert Smolka
Dozent/inn/en	Prof. Dr. Gert Smolka, Prof. Dr.-Ing. Holger Hermanns
Zuordnung zum Curriculum	Bachelor Informatik, Pflicht
Zulassungsvoraussetzungen	Keine
Leistungskontrollen / Prüfungen	• zwei Klausuren (Mitte und Ende der Vorlesungszeit) • Die Note wird aus den Klausuren gemittelt und kann durch Leistungen in den Übungen verbessert werden. • Eine Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.
Lehrveranstaltungen / SWS	Vorlesung: 4 SWS (ca. 250 Studierende) Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 80 h Präsenz- und 190 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

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Programmierung 2					**CS 220 / P2**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
2	6	Jährlich / SS	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Sebastian Hack
Dozent/inn/en	Prof. Dr. Sebastian Hack, Prof. Dr. Andreas Zeller
Zuordnung zum Curriculum	Bachelor Informatik, Pflicht
Zulassungsvoraussetzungen	Programmierung 1 (empfohlen)
Leistungskontrollen / Prüfungen	Prüfungsleistungen werden in zwei Teilen erbracht, die zu gleichen Teilen in die Endnote eingehen. Um die Gesamtveranstaltung zu bestehen, muss jeder Teil einzeln bestanden werden. Im **Praktikumsteil** müssen die Studierenden eine Reihe von Programmieraufgaben selbstständig implementieren. Diese Programmieraufgaben ermöglichen das Einüben der Sprachkonzepte und führen außerdem komplexere Algorithmen und Datenstrukturen ein. Automatische Tests prüfen die Qualität der Implementierungen. Die Note des Praktikumsteils wird maßgeblich durch die Testergebnisse bestimmt. Im **Vorlesungsteil** müssen die Studierenden Klausuren absolvieren und Übungsaufgaben bearbeiten. Die Aufgaben vertiefen dabei den Stoff der Vorlesung. Die Zulassung zu der Klausur hängt von der erfolgreichen Bearbeitung der Übungsaufgaben ab. Im Praktikumsteil kann eine Nachaufgabe angeboten werden
Lehrveranstaltungen / SWS	Vorlesung: 4 SWS Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 45 h Präsenz- und 225 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 Grundprinzipien der imperativen /objektorientierten Programmierung kennen. Dabei wird primär Java als Programmiersprache verwendet.

In dieser Vorlesung lernen sie:

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

Inhalt

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

sowie spezifische Vorlesungen für die Programmieraufgaben.

Weitere Informationen

Unterrichtssprache: deutsch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Empfohlene Vorkenntnisse:

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Systemarchitektur					**CS 230 / SysArch**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
2	6	Jährlich / SS	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. W.-J. Paul
Dozent/inn/en	Prof. Dr. W.-J. Paul Prof. Dr. Jan Reineke
Zuordnung zum Curriculum	Bachelor Informatik, Pflicht
Zulassungsvoraussetzungen	Programmierung 1 und Mathematik für Informatiker 1 (empfohlen)
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. Prüfungsleistungen: erfolgreiche Bearbeitung von 50 % der Übungsaufgaben berechtigt zur Teilnahme an den Klausuren. Bestehen von zwei aus drei Klausuren.
Lehrveranstaltungen / SWS	Vorlesung: 4 SWS Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 80 h Präsenz- und 190 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 die Funktionsweise, die Eigenschaften und die Entwurfsprinzipien von Rechnerarchitekturen und Betriebssystemen kennen lernen.

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. Betriebssystemkern

a. Virtualisierung
b. Ressourcen-Verwaltung, Speicher, Prozessor
c. Scheduling
d. Datei-System

Weitere Informationen

Unterrichtssprache: deutsch

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

Empfohlene Vorkenntnisse:

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Grundzüge von Algorithmen und Datenstrukturen					**CS 340 / GrADS**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
3	6	Jährlich / WS	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	Bachelor Informatik, Pflicht
Zulassungsvoraussetzungen	Programmierung 1 und 2 (empfohlen) Mathematik für Informatiker 1 und 2 (empfohlen)
Leistungskontrollen / Prüfungen	Erfolgreiche Bearbeitung der Übungsblätter berechtigt zur Klausurteilnahme.
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 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

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

Empfohlene Vorkenntnisse:

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Informationssysteme					**CS 330 / InfoSys**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
4	6	Jährlich / SS	1 Semester	4	6

Modulverantwortliche/r	Prof. Dr. Jens Dittrich
Dozent/inn/en	Prof. Dr. Jens Dittrich u.a.
Zuordnung zum Curriculum	4. Semester / Pflicht
Zulassungsvoraussetzungen	Programmierung 1, Programmierung 2, Mathematik für Informatiker 1 sowie Grundzüge Algorithmen und Datenstrukturen
Leistungskontrollen / Prüfungen	Erfolgreiche Teilnahme an den Übungen/Projekt berechtigt zur Teilnahme an der Abschlußklausur (bzw. Studienarbeit).
Lehrveranstaltungen / SWS	Vorlesung: 3 SWS Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	180 h = 80 h Präsenz- und 100 h Eigenstudium
Modulnote	Wird aus Leistungen in Klausuren (alternativ Studienarbeit), Übungen, ggf. Projekt ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekanntgegeben.

Lernziele / Kompetenzen

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

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

UNIVERSITÄT DES SAARLANDES

Themen sind u.a.:

1 Einführung und Einordnung
- 1.1 Einordnung und Abgrenzung: Data Science
- 1.2 Wert von Daten: Das ~~Gold~~ Uran des 21. Jahrhunderts
- 1.3 Bedeutung von Datenbanksystemen
- 1.4 Datenbanksystem vs Dateisystem
- 1.5 Architekturen: 2-Tier, 3-Tier, etc
- 1.6 Daten vs Datenrepräsentation
- 1.7 Datenunabhängigkeit und andere Indirektionen
- 1.8 Modellierung vs Realität
- 1.9 Kosten mangelhafter Modellierung
- 1.10 Datenbanksystem nutzen vs selbst Funktionalität implementieren
- 1.11 Positive Beispiele für Anwendungen
- 1.12 Negative Beispiele für Anwendungen
- 1.13 Anforderungen an Datenbanksysteme
- 1.14 Literaturhinweise

2 Datenmodellierung
- 2.1 Motivation
- 2.2 Übersicht über die Modellierungsschritte
- 2.3 Entity Relationship-Modellierung (E/R)
- 2.4 UML
- 2.5 Relationales Modell
- 2.6 Hierarchische Daten
- 2.7 Graphen und RDF
- 2.8 Key/Value-Modell
- 2.9 Objektrelationale Mapper

3 Anfragesprachen
- 3.1 Relationale Algebra
- 3.2 Hierarchische Daten
- 3.3 Graphen und RDF

4 SQL
- 4.1 SQL Standards und Teilsprachen
- 4.2 Grundlagen
- 4.3 ORDER BY
- 4.4 LIMIT
- 4.5 LIKE
- 4.6 Binäre Operatoren
- 4.7 Joins
- 4.8 Gruppierung und Aggregation
- 4.9 Sichten
- 4.10 ACID

5 Implementierungstechniken
- 5.1 Übersicht
- 5.2 vom WAS zum WIE
- 5.3 Kosten verschiedener Operationen
- 5.4 EXPLAIN
- 5.5 Physisches Design
- 5.6 Indexe, Index-Selection
- 5.7 Datenbank-Tuning
- 5.8 Anfrageoptimierung

6 Zeit und Raum
- 6.1 als Teil des Schemas
- 6.2 as of
- 6.3 append-only und Streaming
- 6.4 Versioning

- 6.5 Snapshotting
- 6.6 Differential Files
- 6.7 Publish/Subscribe
- 6.8 Indexstrukturen

7 Recovery, Durability, Archivierung
- 7.1 Grundproblematik
- 7.2 Vergessen vs Komprimieren vs Kondensieren
- 7.3 Heiße vs kalte Daten
- 7.4 Archivierung
- 7.5 Redundanz
- 7.6 Implementierungsaspekte
- 7.7 UNDO/REDO
- 7.8 Logging

8 Nebenläufigkeitskontrolle
- 8.1 Isolationlevels
- 8.2 Eventual Consistency
- 8.3 2PL
- 8.4 Verteilte Datenbanksysteme: Grundkonzepte: Sharding, HP; VP, QP
- 8.5 Implementierungsaspekte

9 ETL
- 9.1 Datenbankschnittstellen: JDBC et al
- 9.2 Textdatenbanken: NoDB, CSV
- 9.3 Föderierte Datenbanken
- 9.4 Data Warehousing
- 9.5 Schema Matching
- 9.6 Reporting
- 9.7 Data Cleaning
- 9.8 Redundanz und Normalisierung
- 9.9 Denormalisierung
- 9.10 Workflows

10 Big Data
- 10.1 Was ist eigentlich Big Data?
- 10.2 Big Data vs Privatheit
- 10.3 Beispiele: Zusammenführen von Daten
- 10.4 Physische Barrieren

11 NoSQL
- 11.1 Key/Value Stores
- 11.2 KeyDocument Stores: MongoDB
- 11.3 MapReduce
- 11.4 Flink
- 11.5 Spark

12 Information Retrieval
- 12.1 Inverted Files
- 12.2 Stemming
- 12.3 Ranking

13 Potpourri
- 13.1 Deduktive DBMS
- 13.2 Probabilistische DBMS

Weitere Informationen

Unterrichtssprache: Deutsch

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

Empfohlene Vorkenntnisse:
Softwarepraktikum

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Nebenläufige Programmierung					**CS 430**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
4	6	Jährlich / SS	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, Ph.D Prof. Dr. Verena Wolf
Zuordnung zum Curriculum	Bachelor Informatik, Pflicht
Leistungskontrollen / Prüfungen	Zwei Klausuren (Mitte und Ende der Vorlesungszeit), praktisches Projekt. Nachklausuren finden innerhalb der letzten Wochen vor Vorlesungsbeginn des Folgesemesters statt..
Zulassungsvoraussetzungen	Programmierung 1 und 2, Softwarepraktikum, Grundzüge der Theoretischen Informatik (empfohlen)
Modulelemente / SWS	**Element T – Theorie (2 SWS)**: 8 Vorlesungen: 6 Wochen (ca. 150 Studierende) 4 Übungen: 6 Wochen (Übungsgruppen mit ca. 20 Studierenden) **Element A – Anwendung (2 SWS)**: 9 Vorlesungen: 6 Wochen (ca. 150 Studierende) 4 Übungen: 6 Wochen (Übungsgruppen mit ca. 20 Studierenden) **Element P – Praxis (2 SWS)**: Semesterbegleitend 8 schriftliche Reflektionen (Prüfungsvorleistungen), anschließend Projektarbeit über ca. 2 Wochen
Arbeitsaufwand	**Element T**: 24 h Präsenz, 36h Selbststudium **Element A**: 26 h Präsenz, 34h Selbststudium **Element P**: 60 h Selbststudium
Modulnote	Wird aus Leistungen in Klausuren (im Anschluss an die Elemente T und A), sowie den Prüfungsvorleistungen (Element P) ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben. Alle Modulelemente sind innerhalb eines Prüfungszeitraumes erfolgreich zu absolvieren.

Lernziele / Kompetenzen

Die Teilnehmer lernen die Nebenläufigkeit von Prozessen als ein weitreichendes, grundlegendes Prinzip in der Theorie und Praxis der modernen Informatik kennen. Durch die Untersuchung und

Verwendung unterschiedlicher formaler Modelle gewinnen die Teilnehmer ein vertieftes Verständnis von Nebenläufigkeit. Dabei lernen die Teilnehmer wichtige formale Konzepte der Informatik korrekt anzuwenden. Das im ersten Teil der Veranstaltung erworbene theoretische Wissen wird in der zweiten Hälfte in der (Programmier-) Praxis angewendet. Dabei lernen die Teilnehmer verschiedene Phänomene des nebenläufigenden Programmierens in den formalen Modellen zu beschreiben und mit deren Hilfe konkrete Lösungen für die Praxis abzuleiten. Desweiteren werden die Teilnehmer in der Praxis existierende Konzepte auf diese Art auf ihre Verlässlichkeit hin untersuchen.

Inhalt

Nebenläufigkeit als Konzept
- -- Potentieller Parallelismus
- -- Tatsächlicher Parallelismus
- -- Konzeptioneller Parallelismus

Nebenläufigkeit in der Praxis
- -- Objektorientierung
- -- Betriebssysteme
- -- Multi-core Prozessoren, Coprozessoren
- -- Programmierte Parallelitaet
- -- Verteilte Systeme
  (client-server, peer-2-peer, Datenbanken, Internet)

Die Schwierigkeit von Nebenläufigkeit
- -- Ressourcenkonflikte
- -- Fairness
- -- Gegenseitiger Auschluss
- -- Verklemmung (Deadlock)
- -- gegenseitige Blockaden (Livelock)
- -- Verhungern (Starvation)

Grundlagen der Nebenläufigkeit
- -- Sequentielle Prozesse
- -- Zustaende, 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
- -- pseuCo
- -- Message-passing concurrency
- -- Message-passing in pseuCo und Go
- -- Shared-memory concurrency
- -- Monitore und Semaphoren
- -- Shared-memory in pseuCo und Java
- -- Shared Objects und Threads in Java
- -- Shared Objects und Threads als Transitionssysteme

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

Weitere Informationen

Unterrichtssprache: deutsch

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

Empfohlene Vorkenntnisse:
Programmierung 1 und 2, Softwarepraktikum, Grundzüge der Theoretischen Informatik

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Grundzüge der Theoretischen Informatik					**CS 420 / TheoInf**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
3	6	Jährlich / WS	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, Prof. Dr. Markus Bläser
Zuordnung zum Curriculum	Bachelor Informatik, Pflicht
Zulassungsvoraussetzungen	Programmierung 1 und 2, Mathematik für Informatiker 1 und 2 (empfohlen)
Leistungskontrollen / Prüfungen	Erfolgreiche Bearbeitung der Übungsaufgaben berechtigt zur Klausurteilnahme.
Lehrveranstaltungen / SWS	Vorlesung 4 SWS Übung 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand	270 h = 80 h Präsenz- und 190 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

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

Empfohlene Vorkenntnisse:

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Softwaredesignpraktikum					**CS 320 / SoDePra**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
3	6	Vorlesungsfreie Zeit nach dem SS	6 Wochen	6	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 Bachelor Informatik, Pflicht

Zulassungsvoraussetzungen Programmierung 1 und 2 (empfohlen)

Leistungskontrollen / Prüfungen

- Erfolgreiches Erstellen im Team eines komplexen Software-Produkts, insbesondere
- Einreichen der erforderlichen Dokumente
- Abnahme des Endprodukts durch den Kunden
- Einhaltung der Termin- und Qualitätsstandards

Lehrveranstaltungen / SWS Vorlesung 2 SWS
Praktikum 4 SWS (Teams in Gruppen bis zu 6 Studierende)

Arbeitsaufwand 270 h = 20 h Präsenz- und 250 h Eigenstudium

Modulnote unbenotet

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 früher erkannt 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.

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

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Operating Systems, Core Course					**CS 551 / OS**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortlicher	Prof. Peter Druschel, Ph.D.
Dozent	Prof. Peter Druschel, Ph.D. Björn Brandenburg, Ph.D.
Zuordnung zum Curriculum	Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: none
Leistungskontrollen / Prüfungen	Regular attendance at classes and tutorials Successful completion of a course project in teams of 2 students Passing 2 written exams (midterm and final exam) A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Lehrveranstaltungen /SWS	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

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

Inhalt

Process management:
- Threads and processes, synchronization
- Multiprogramming, CPU Scheduling
- Deadlock

Memory management:
- Dynamic storage allocation
- Sharing main memory
- Virtual memory

I/O management:
- File storage management
- Naming
- Concurrency, Robustness, Performance

Virtual machines

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesusng auf der Vorlesungsseite im Internet

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Computer Graphics, Core Course					**CS 552 / CG**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortlicher	Prof. Dr. Philipp Slusallek
Dozent	Prof. Dr. Philipp Slusallek
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: none
Leistungskontrollen / Prüfungen	• Sucessful completion of at least 50% of the exercises • Sucessful 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 and use it in a high-quality rendering.

Inhalt

- Fundamentals of digital image synthesis
  - Phyical 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: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesusng auf der Vorlesungsseite im Internet

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Database Systems, Core Course					**CS 553 / DBS**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortlicher Prof. Dr. Jens Dittrich

Dozent Prof. Dr. Jens Dittrich

Zuordnung zum Curriculum Graduate course / Mandatory Elective

Zulassungsvoraussetzungen

especially Saarland University CS department's undergraduate lecture "Informationssysteme", "Prog 1", "Prog 2", "Algorithmen und Datenstrukturen" as well as "Nebenläufige Programmierung"

For graduate students:

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

Leistungskontrollen / Prüfungen

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

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

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

Lehrveranstaltungen / SWS	Lecture 4 h (weekly; this class may be run as a flipped classroom, i.e. 2 hours may be replaced by self-study of videos/papers; the other 2 hours may be used to run a group exercice supervised by the professor called “the LAB”) 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	Will be determined based on project, midterm and best of endterm and reexam.

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 lifes -- would be impossible in their present forms. The course teaches the architectural and algorithmic foundations of modern database management systems (DBMS), focussing on database systems internals rather than applications. Emphasis is made on robust and time-tested techniques that have led databases to be considered a mature technology and one of the greatest success stories in computer science. At the same time, opportunities for exciting research in this field will be pointed out.

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

Inhalt

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

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

- write-optimized system concepts (OLTP, streaming data)
- management of geographical data (GIS, google maps and similar tools)
- main-memory techniques

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Embedded Systems, Core Course					**CS 650 /ES**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Bernd Finkbeiner, Ph.D
Dozent/inn/en	Prof. Bernd Finkbeiner, Ph.D
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	
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) The course is accompanied by a laboratory project, in which a non-trivial embedded system has to be realized.
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: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Information Retrieval and Data Mining, Core Course					**CS 555 / IRDM**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Gerhard Weikum
Dozent/inn/en	Prof. Dr. Gerhard Weikum
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	Good knowledge of undergraduate mathematics (linear algebra, probability theory) and basic algorithms.
Leistungskontrollen / Prüfungen	• Regular attendance of classes and tutor groups • Presentation of solutions in tutor groups • Passing 2 of 3 written tests (after each third of the semester) • Passing the final exam (at the end of the semester)
Lehrveranstaltungen / SWS	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	Will be determined by the performance in written tests, tutor groups, and the final exam. Details will be announced on the course web site.

Lernziele / Kompetenzen

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

Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise: will be announced on the course web site.

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Artificial Intelligence, Core Course					**CS 556 / AI**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Jörg Hoffmann
Dozent/inn/en	Prof. Dr. Jörg Hoffmann, Prof. Dr. Wolfgang Wahlster
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: none
Leistungskontrollen / Prüfungen	• Regular attendance of classes and tutorials • Solving of weekly assignments • Passing the final written exam • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Lehrveranstaltungen / SWS	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 30 students
Arbeitsaufwand	270 h = 90 h of classes and 180 h private study
Modulnote	Wird aus Leistungen in Klausuren ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

Knowledge about basic methods in Artificial Intelligence

Inhalt

Problem-solving:
- Uninformed- and informed search procedures
- Adversarial search

Knowledge and reasoning:
- Propositional logic
- SAT
- First-order logic, Inference in first-order logic
- Knowledge representation, Semantic Web
- Default logic, rule-based mechanisms

Planning:
- STRIPS formalism and complexity
- Delete relaxation heuristics

Probabilistic reasoning:
- Basic probabilistic methods
- Bayesian networks

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Russel & Norvig „Artificial Intelligence: A Modern Approach“; weitere ggf. per Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Computer Architecture, Core Course					**CS 558 / CAR**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. W.-J. Paul
Dozent/inn/en	Prof. Dr. W.-J. Paul
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: none
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 at the end of the 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 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.

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: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Security, Core Course					**CS 559 / SEC**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Matteo Maffei
Dozent/inn/en	Prof. Dr. Michael Backes, Prof. Dr. Christian Hammer, Prof. Dr. Matteo Maffei
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: none
Leistungskontrollen / Prüfungen	• Regular attendance of classes and tutorials • Passing the final exam • A re-exam is normally provided (as written or oral examination).
Lehrveranstaltungen / SWS	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 will acquire a deep understanding and hands-on experience on a broad spectrum of attack and defense techniques for IT systems.

Inhalt

- Security principles
- Authentication and access control
- Network security
- Operating system security
- Web application security
- Malware
- Risk management
- Logging and log analysis
- Cryptographic protocols
- Security of information flow

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Software Engineering, Core Course					CS 560 / SE
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Andreas Zeller
Dozent/inn/en	Prof. Dr. Andreas Zeller
Zuordnung zum Curriculum	Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: none
Leistungskontrollen / Prüfungen	• Successful project completion (including deliverables such as requirements, design, implementation) • Successful project demonstration • Regular attendance of classes • Passing the final exam
Lehrveranstaltungen / SWS	Lecture 2 h (weekly) Project 4 h (weekly) Project work in teams of 4–7 students
Arbeitsaufwand	270 h = 90 h of classes and 180 h private study
Modulnote	Wird aus Leistungen in Klausuren 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 systematic elicitation of requirements and how to document them.

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.

They apply these techniques in a project in small teams.

Lecture Inhalts

- 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: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Compiler Construction, Core Course					**CS 561 / CC**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r — Prof. Dr. Sebastian Hack

Dozent/inn/en — Prof. Dr. Sebastian Hack

Zuordnung zum Curriculum — Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen — For graduate students: none

Leistungskontrollen / Prüfungen

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

Lehrveranstaltungen / SWS — 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, foundations of program analysis.

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Automated Reasoning					**CS 571 / AR**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Christoph Weidenbach
Dozent/inn/en	Prof. Dr. Christoph Weidenbach
Zuordnung zum Curriculum	Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	CS 575 ICL
Leistungskontrollen / Prüfungen	• Regular attendance of classes and tutorials • Weekly assignments • Practical work with systems • Passing the final and mid-term exam • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Lehrveranstaltungen / SWS	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 logics, calculi, implementation techniques, and systems providing automated reasoning.

Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Image Processing and Computer Vision, Core Course					**CS 572 / IPCV**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Joachim Weickert
Dozent/inn/en	Prof. Dr. Joachim Weickert
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: none
Leistungskontrollen / Prüfungen	• Regular attendance of classes and tutorials. • At least 50% of all possible points from the weekly assignments have to be gained to qualify fort he final exam. • 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

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

Inhalt

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-Smoothers
   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

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Computer Algebra, Core Course					**CS 573 / CA**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Frank-Olaf Schreyer
Dozent/inn/en	Prof. Dr. Frank-Olaf Schreyer
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: none
Leistungskontrollen / Prüfungen	• Regular attendance of classes and tutorials • Solving the exercises, passing the midterm and the final exam.
Lehrveranstaltungen / SWS	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

Solving problems occuring in computer algebra praxis
The theory behind algorithms

Inhalt

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

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

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Algorithms and Data Structures, Core Course					**CS 574 / A&D**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Kurt Mehlhorn
Dozent/inn/en	Prof. Dr. Kurt Mehlhorn, Prof. Dr. Raimund Seidel
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: C, C++, Java
Leistungskontrollen / Prüfungen	• Regular attendance of classes and tutorials • Passing the midterm and the final exam • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Lehrveranstaltungen / SWS	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 standard algorithms for typical problems in the areas graphs, computational geometry, strings and optimization. Furthermore they master a number of methods and data-structures to develop efficient algorithms and analyze their running times.

Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Introduction to Computational Logic, Core Course					**CS 575 / ICL**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Gert Smolka

Dozent/inn/en Prof. Dr. Gert Smolka

Zuordnung zum Curriculum
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen

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

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

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

Inhalt

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

Predicate Logic (higher-order)
- quantifier axioms
- natural deduction
- prenex and Skolem forms

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Geometric Modeling, Core Course					**CS 576 / GM**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	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 — Graduate course / Mandatory Elective

Zulassungsvoraussetzungen — For graduate students: none

Leistungskontrollen / Prüfungen

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

Lehrveranstaltungen / SWS

Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students (theory)
Practical assignments in groups of 3 students (practice)
Tutorials consists of a mix of theoretical + practical assignments.

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

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

Lernziele / Kompetenzen

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

Inhalt

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

Weitere Informationen

Unterrichtssprache: English

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Complexity Theory, Core Course					**CS 577 / CT**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Markus Bläser
Dozent/inn/en	Prof. Dr. Markus Bläser, Prof. Dr. Raimund Seidel
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	none undergraduate course on theory of computation (e.g. "Grundzüge der Theoretischen Informatik“) is highly recommend.
Leistungskontrollen / Prüfungen	• Regular attendance of classes and tutorials • assignments • exams (written or oral)
Lehrveranstaltungen / SWS	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of about 20 students
Arbeitsaufwand	270 h = 90 h of classes and 180 h private study
Modulnote	Will be calculated from the results in the assignments and/or exams, as announced by the Lecturer at the beginning of the course

Lernziele / Kompetenzen

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

Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Cryptography, Core Course					**CS 578 / CRY**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Michael Backes
Dozent/inn/en	Prof. Dr. Michael Backes, Prof. Dr. Dominique Schröder
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: Basic knowledge in theoretical computer science required, background knowledge in number theory and complexity theory helpful
Leistungskontrollen / Prüfungen	• Oral / written exam (depending on the number of students) • A re-exam is normally provided (as written or oral examination).
Lehrveranstaltungen / SWS	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 will acquire a comprehensive knowledge of the basic concepts of cryptography and formal definitions. They will be able to prove the security of basic techniques.

Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Optimization, Core Course					**CS 579 / OPT**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r Prof. Dr. Kurt Mehlhorn

Dozent/inn/en Prof. Dr. Kurt Mehlhorn

Zuordnung zum Curriculum
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

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

Lehrveranstaltungen / SWS
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 to model and solve optimization problems from theory as from the real world

Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Semantics, Core Course					**CS 580 / SEM**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Gert Smolka
Dozent/inn/en	Prof. Dr. Gert Smolka
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: core lecture Introduction to Computational Logic
Leistungskontrollen / Prüfungen	• Regular attendance of classes and tutorials. • Passing the midterm and the final exam
Lehrveranstaltungen / SWS	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

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

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Verification, Core Course					**CS 581 / VERI**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Holger Hermanns
Dozent/inn/en	Prof. Dr. Holger Hermanns, Prof. Bernd Finkbeiner, Ph.D
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: none
Leistungskontrollen / Prüfungen	• Regular attendance of classes and tutorials • Passing the final exam • A re-exam takes place during the last two weeks before the start of lectures in the following semester. .
Lehrveranstaltungen / SWS	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 become familiar with the standard methods in computer-aided verification. They understand the theoretical foundations and are able to assess the advantages and disadvantages of different methods for a specific verification project.
The students gain first experience with manual correctness proofs and with the use of verification tools.

Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Telecommunications I, Core Course					**TC I**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr.-Ing. Thorsten Herfet
Dozent/inn/en	Prof. Dr.-Ing. Thorsten Herfet
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	The lecture requires a solid foundation of mathematics (differential and integral calculus) and probability theory. The course will, however, refresh those areas indispensably necessary for telecommunications and potential intensification courses and by this open this potential field of intensification to everyone of you.
Leistungskontrollen / Prüfungen	Regular attendance of classes and tutorials Passing the final exam in the 2nd week after the end of courses. Eligibility: Weekly exercises / task sheets, grouped into two blocks corresponding to first and second half of the lecture. Students must provide min. 50% grade in each of the two blocks to be eligible for the exam.
Lehrveranstaltungen / SWS	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	Final exam mark

Lernziele / Kompetenzen

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

Inhalt

As the basic principle, the course will give an introduction into the various building blocks that modern telecommu¬nication systems do incorporate. Sources, sinks, source and channel coding, modulation and multiplexing are the major keywords but we will also deal with dedicated pie¬ces like A/D- and D/A-converters and quantizers in a little bit more depth.
The course will refresh the basic transformations (Fou¬rier, Laplace) that give access to system analysis in the frequency domain, it will introduce derived transformations (z, Hilbert) for the analysis of discrete systems and modulation schemes and it will briefly introduce algebra on finite fields to systematically deal with error correction schemes that play an important role in modern communi¬cation systems.

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Machine Learning, Core Course					**ML**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Matthias Hein
Dozent/inn/en	Prof. Dr. Matthias Hein
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	The lecture gives a broad introduction into machine learning methods. After the lecture the students should be able to solve and analyze learning problems.
Leistungskontrollen / Prüfungen	• Regular attendance of classes and tutorials. • 50% of all points of the exercises have to be obtained in order to qualify for the exam. • Passing 1 out of 2 exams (final, re-exam).
Lehrveranstaltungen / SWS	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	Determined from the results of the exams, exercises and potential projects. The exact grading modalities are announced at the beginning of the course.

Lernziele / Kompetenzen

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

Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Distributed Systems, Core Course					**DS**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Peter Druschel, Ph.D.
Dozent/inn/en	Prof. Peter Druschel, Ph.D. Allen Clement, Ph.D.
Zuordnung zum Curriculum	Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	Operating systems or concurrent programming.
Leistungskontrollen / Prüfungen	• Regular attendance at classes and tutorials. • Successful completion of a course project in teams of 2 students. (Project assignments due approximately every 2 weeks.) • Passing grade on 2 out of 3 written exams: midterm, final exam, and a re-exam that takes place during the last two weeks before the start of lectures in the following semester. • Final course grade: 50% project, 50% best 2 out of 3 exams.
Lehrveranstaltungen / SWS	Lecture 4 h (weekly) Tutorial 2 h (weekly)
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

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

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

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Data Networks, Core Course					**CS 554 / DN**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Holger Hermanns
Dozent/inn/en	Prof. Dr. Holger Hermanns
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	For graduate students: none
Leistungskontrollen / Prüfungen	• Regular attendance of classes and tutorials • Qualification for final exam through mini quizzes during classes • Possibility to get bonus points through excellent homework • Final exam • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Lehrveranstaltungen / SWS	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: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Computer Architecture 2, Advanced Course					CS 650 / CAR2
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
6	5 - 6	At least once every two years	1 Semester	6	9

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

Dozent/inn/en — Prof. Dr. W. J. Paul

Zuordnung zum Curriculum — Bachelor Informatik
Master Informatik
Graduate / Mandatory Elective

Zulassungsvoraussetzungen — Related core lecture Computer Architecture

Leistungskontrollen / Prüfungen — Studying:
Students should to 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 floationg 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: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Telecommunications II, Advanced Course					CS 650 / TC II
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
6	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr.-Ing. Thorsten Herfet
Dozent/inn/en	Prof. Dr.-Ing. Thorsten Herfet
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	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
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	Final Exam Mark

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 802.11n and the terrestrial DVB system (DVB-T2).

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Automata, Games and Verification, Advanced Course					CS 650 / AG&V
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
6	5 - 6	At least once every two years	1 Semester	4	6

Modulverantwortliche/r Prof. Bernd Finkbeiner, PhD

Dozent/inn/en Prof. Bernd Finkbeiner, PhD

Zuordnung zum Curriculum
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen

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: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Automated Debugging, Advanced Course					CS 650 / AutoD
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
6	5 - 6	At least once every two years	1 Semester	4	6

Modulverantwortliche/r	Prof. Dr. Andreas Zeller
Dozent/inn/en	Prof. Dr. Andreas Zeller
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate / Mandatory Elective
Zulassungsvoraussetzungen	Programming skills as acquired at the Bachelor level
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: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Computer Graphics II, Advanced Course Realistic Image Synthesis					**CS 650 / CGII-RIS**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Philipp Slusallek
Dozent/inn/en	Prof. Dr. Philipp Slusallek, Dr. Karol Myszkowski
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	Related core lecture Computer Graphics
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	Will be determined by the performance in written tests, tutor groups, and the final exam. Details Bekanntgabe jeweils vor Beginn der Vorlesusng auf der Vorlesungsseite im Internet

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
- Theory of Variance Reduction
- Direct Illumination, Importance Sampling
- BRDF, Inversion Methods
- Distribution Ray Tracing and Path Tracing
- Bidirectional Path Tracing, Instant Radiosity
- Density Estimation Methods, Photon Mapping
- Advanced Lighting Simulation Algorithms
- Rendering of Animations
- Motion Blur, Temporal Filtering
- Interactive Global Illumination
- Hardware Rendering Basics
- Advanced Hardware Rendering
- Tone Mapping, Perception

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Differential Equations in Image Processing and Computer Vision, Advanced Course					**CS 650 / DIC**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
6	5 - 6	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Joachim Weickert
Dozent/inn/en	Prof. Dr. Joachim Weickert
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Zulassungsvoraussetzungen	Related core lecture Computer Vision
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 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 Semidiscete 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

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Introduction to Image Acquisition Methods, Advanced Course					**CS 750 / IIAM**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
6	5 - 6	At least once every two years	1 Semester	2	4

Modulverantwortliche/r	Prof. Dr. Joachim Weickert
Dozent/inn/en	N. N.
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate Course / Elective
Zulassungsvoraussetzungen	Related core lecture Computer Vision
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)
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

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Correspondence Problems in Computer Vision, Advanced Course					**COPCV**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
6	5 - 6	At least once every two years	1 Semester	4	6

Modulverantwortliche/r	Prof. Dr. Joachim Weickert
Dozent/inn/en	Prof. Dr. Joachim Weickert
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Graduate Course / Elective
Zulassungsvoraussetzungen	Related core lecture Computer Vision, Completed Mathematics for Computer Scientist lectures.
Leistungskontrollen / Prüfungen	- Regular attendence of lecture and tutorial - Written or oral exam and the end of the course
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

Correspondence problems are a central topic in computer vision. Thereby, one is interested in identifying and matching corresponding features in different images/views of the same scene. Typical corresondence problems are the estimation of motion information from consecutive frames of an image sequence (optic flow), the reconstruction of a 3-D scene from a stereo image pair and the registration of medical image data from different modalities (e.g. CT and MRT).
Central part of this lecture is the discussion of the most important correspondence problems as well as the modelling of suitable algorithms for solving them.

Inhalt

1. Introduction and Overview
2. General Matching Concepts
   2.1 Block Matching
   2.2 Correlation Techniques
   2.3 Interest Points
   2.4 Feature-Based Methods
3. Optic Flow I
   3.1 Local Differential Methods
   3.2 Parameterisation Models
4. Optic Flow II
   4.1 Global Differential Methods
   4.2 Horn and Schunck
5. Optic Flow III
   5.1 Advanced Constancy Assumptions
   5.2 Large Motion
6. Optic Flow IV
   6.1 Robust Data Terms
   6.2 Discontinuity-Preserving Smoothness Terms
7. Optic Flow V
   7.1 High Accuracy Methods
   7.2 SOR and Lienar Multigrid
8. Stereo Matching I
   8.1 Projective Geometry
   8.2 Epipolar Geometry
9. Stereo Matching II
   9.1 Estimation of the Fundamental Matrix
10. Stereo Matching III
    10.1 Correlation Methods
    10.2 Variational Approaches
    10.3 Graph Cuts
11. Medical Image Registration
    11.1 Mutual Information
    11.2 Elastic and Curvature Based Registration
    11.3 Landmarks
12. Particle Image Velocimetry
    12.1 Div-Curl-Regularisation
    12.2 Incompressible Navier Stokes Prior

Weitere Informationen

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Future Media Internet – FMI, Advanced Cours					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
6	5-6	WS	1 Semester	3	9

Modulverantwortliche/r	Prof. Dr. Jörg Hoffmann
Dozent/inn/en	Prof. Dr. Jörg Hoffmann
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Extended Courses
Zulassungsvoraussetzungen	For graduate students: none
Leistungskontrollen / Prüfungen	Regular attendance of classes and tutorial. Paper as well as programming exercises for exam qualification Final exam A re-exam takes place before the start of lectures in the following semester
Lehrveranstaltungen / SWS	Lecture 3 h (weekly) Tutorial 1 h (weekly)
Arbeitsaufwand	270 h = 60 h classes, 90 h private study, 120 h programming exercises
Modulnote	Graded absolute 1.0-n.b. and relative A-F

Lernziele / Kompetenzen

The course deals with Media Transport over the Internet. After the course students know how data- and mediatransport is solved in today's Internet and have a good understanding of so called erasure channels.
Besides the pure transport protocol design the course complements the fundaments laid in TCI and TCII be introducing state-of-the-art error codes (Van-der-Monde-Codes, Fountain Codes) and by engineering tasks like the design of a Digital PLL.

Inhalt

The course introduces media transmission over packet channels, specifically the Internet. After establishing a Quality of Service framework built on ITU requirements the course models erasure channels without and with memory. Key characteristics like the channel capacity and the minimum redundancy information are derived.

The second part of the course introduces current media transport protocol suites (TCP, UDP, RTP, RTSP) and middleware (ISMA, DLNA, UPnP, DVB-IPI).

In the second half of the course audiovisual coders used in the Internet are introduced (H.264, AAC), state-of-the-art forward error coding schems (Van-der-Monde-Codes, Fountain Codes) are explained and essential elements like a Digital Phase-locked Loop are developed.

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
The course will come with a self contained manuscript. The most essential monographs used for and referenced within the manuscript are available in the Computer Science Library of Saarland University.

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Automatic Planning, Advanced Course					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
6	5-6	WS	1 Semester	4	9

Responsible Lecturer	Prof. Dr Jörg Hoffmann
Lecturer	Prof. Dr. Jörg Hoffmann
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	Regular attendance of classes and tutorial Paper as well as programming exercises for exam qualification Final exam A re-exam takes place before the start of lectures in the following semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly)
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Aims / Competences to be developed

The students will gain a deep understanding of algorithms used in Automatic Planning for the efficient exploration of large state spaces, from both a theoretical and practical point of view. The programming exercises will familiarize them with the main implementation basis in Automatic Planning. The search algorithms are generic and are relevant also in other CS sub-areas in which large transition systems need to be analyzed.

Content

Automatic Planning is one of the fundamental sub-areas of Artificial Intelligence, concerned with algorithms that can generate strategies of action for arbitrary autonomous agents in arbitrary environments. The course examines the technical core of the current research on solving this kind of problem, consisting of paradigms for automatically generating heuristic functions (lower bound solution cost estimators), as well as optimality-preserving pruning methods. Apart from understanding these techniques themselves, the course explains how to analyze, combine, and compare them.

Starting from an implementation basis provided, students implement their own planning system as part of the course. The course is concluded by a competition between these student systems.

Additional Information

Unterrichtssprache: Englisch

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

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Modul Proseminar Unterschiedliche Themen					**CS 300**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
4	4 - 6	Jedes Semester		2	5

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

Dozent/inn/en — Professoren der Fachrichtung

Zuordnung zum Curriculum — Bachelor Informatik, Pflicht

Zulassungsvoraussetzungen — Keine

Leistungskontrollen / Prüfungen

- Diskussion in der Gruppe
- thematischer Vortrag
- kurze schriftliche Ausarbeitung

Lehrveranstaltungen / SWS — Proseminar 2 SWS (bis zu 20 Studierende)

Arbeitsaufwand — 150h = 40 h Präsenz und 110 h Eigenstudium

Modulnote — Die Modalitäten der Notenvergabe werden vom verantwortlichen Hochschullehrer festgelegt.

Lernziele / Kompetenzen

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

Sie haben Kompetenz im Verstehen einfacher wissenschaftlicher Aufsätze und im Präsentieren von wissenschaftlichen Erkenntnissen erworben.

Inhalt

Unter Anleitung werden folgende Punkte praktisch geübt:

- Lesen und Verstehen wissenschaftlicher Aufsätze
- Diskutieren der Aufsätze in der Gruppe
- Analysieren, Zusammenfassen und Wiedergeben des spezifischen Themas
- Präsentationstechnik
- Spezifische Vertiefung in Bezug auf das individuelle Thema des Seminars.

Weitere Informationen

Unterrichtssprache: deutsch

Literaturhinweise:
dem Thema entsprechend

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Modul Seminar Changing Topics					**CS 500**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
5	5 - 6	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 — Bachelor Informatik
Master Informatik
Pflicht

Zulassungsvoraussetzungen — Grundlegende Kenntnisse im jeweiligen Teilbereich der Informatik.

Leistungskontrollen / Prüfungen

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

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

Arbeitsaufwand — 210 h = 60 h Präsenz und 150 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 wissenschaftliches 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

Unterrichtssprache: Deutsch/Englisch

Literatur:
dem Thema entsprechend

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Bachelor-Seminar					**CS 690**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
6	6	jedes Semester	1 Semester	5	9

Modulverantwortliche/r	Studiendekan der Fakultät Mathematik und Informatik bzw. Studienbeauftragter der Informatik
Dozent/inn/en	Professoren der Fachrichtung
Zuordnung zum Curriculum	Bachelor Informatik, Pflicht
Zulassungsvoraussetzungen	Gesamter Pflichtkanon des Bachelorstudiengangs, bis auf Bachelorseminar und –arbeit.
Leistungskontrollen / Prüfungen	• Vorstellung eines wissenschaftlichen Artikels im Lesekreis. • Aktive Teilnahme an der Diskussion im Lesekreis. • Vortrag über die geplante Aufgabenstellung mit anschliessender Diskussion. • Schriftliche Beschreibung der Aufgabenstellung der Bachelorarbeit.
Lehrveranstaltungen / SWS	Seminar (Lesekreis) 3 SWS Praktikum 2 SWS
Arbeitsaufwand	280 h = 85 h Präsenz 195 h Selbststudium 3 h pro Woche im Lesekreis 2 h pro Woche Praktikum 13 h pro Woche Selbststudium 5 h direkte Betreuung durch Lehrstuhlmitarbeiter
Modulnote	benotet

Lernziele / Kompetenzen

Im Bachelorseminar erwirbt der Studierende unter Anleitung die Fähigkeit zum wissenschaftlichen Arbeiten im Kontext eines angemessen Themengebietes.

Am Ende des Bachelorseminars sind die Grundlagen für eine erfolgreiche Anfertigung der Bachelorarbeit gelegt, und wesentliche Lösungsansätze bereits eruiert.

Das Bachelorseminar bereitet somit die Themenstellung und Ausführung der Bachelorarbeit vor.

Es vermittelt darüber hinaus praktische Fähigkeiten des wissenschaftlichen Diskurses. Diese Fähigkeiten werden durch die aktive Teilnahme an einem Lesekreis vermittelt, in welchem die Auseinandersetzung mit wissenschaftlich anspruchsvollen Themen geübt wird.

Inhalt

Auf der Grundlage des "state-of-the-art" werden die Methoden der Informatik systematisch unter Anleitung angewendet.

Weitere Informationen

Unterrichtssprache: Deutsch oder Englisch

Literaturhinweise::
Dem Themengebiet entsprechende wissenschaftliche Artikel in enger Absprache mit dem Dozenten

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Bachelorarbeit					**CS 699**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
6	6	jedes Semester	1 Semester		12

Modulverantwortliche/r	Studiendekan der Fakultät Mathematik und Informatik bzw. Studienbeauftragter der Informatik
Dozent/inn/en	Professoren der Fachrichtung
Zuordnung zum Curriculum	Bachelor Informatik Pflicht
Zulassungsvoraussetzungen	Erfolgreicher Abschluss des Bachelor-Seminars
Leistungskontrollen / Prüfungen	Schriftliche Ausarbeitung. Sie beschreibt sowohl das Ergebnis der Arbeit als auch den Weg, der zu dem Ergebnis führte. Der eigene Anteil an den Ergebnissen muss klar erkennbar sein. Außerdem Präsentation der Bachelorarbeit in einem Kolloquium, in dem auch die Eigenständigkeit der Leistung des Studierenden überprüft wird.
Lehrveranstaltungen / SWS	
Arbeitsaufwand	360 h = 20 h Präsenz- und 340 h Eigenstudium
Modulnote	Aus der Beurteilung der Bachelorarbeit

Lernziele / Kompetenzen

Die Bachelor-Arbeit ist eine Projektarbeit, die unter Anleitung ausgeführt wird. Sie zeigt, dass der Kandidat/die Kandidatin in der Lage ist, innerhalb einer vorgegebenen Frist ein Problem aus dem Gebiet der Informatik unter Anleitung zu lösen und die Ergebnisse zu dokumentieren.

Inhalt

Auf der Grundlage des "state-of-the-art" wird die systematische Anwendung der Methoden der Informatik dokumentiert.

Weitere Informationen

Unterrichtssprache: Deutsch / Englisch

Literatur:
Je nach Thema in Absprache mit dem Professor

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Modul Praktikum zum Informationsmanagement					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
		jedes Semester / Beginn jederzeit möglich	1 Semester	4 oder 6	6 oder 9

Modulverantwortliche/r	Prof. Dr. Schmidt
Dozent/inn/en	Prof. Dr. Schmidt und Mitarbeiter
Zuordnung zum Curriculum	Bachelor Informatik Master Informatik Freie Leistungspunkte
Zulassungsvoraussetzungen	Programmierkenntnisse in Java und/oder VB.NET
Leistungskontrollen / Prüfungen	Erstellen eines schriftlichen Berichtes in Paperform (Projektdokumentation) sowie das Bestehen einer mündlichen Prüfung
Lehrveranstaltungen / SWS	Praktikum / 4 SWS oder 6 SWS
Arbeitsaufwand	180 (4 SWS) / 270 (6 SWS) Arbeitsstunden
Modulnote	

Lernziele / Kompetenzen

- Vertiefung von Kenntnissen über Methoden und Modelle des Operations Research
- Transfer von wissenschaftlichen Erkenntnissen in praxistaugliche Lösungen
- Selbständiges Arbeiten an einer Lösung für ein gegebenes Problem (Projektmanagement)

Inhalt

Das Praktikum umfasst wechselnde Aufgabenstellungen aus dem Bereich Operations Research und wird zum größten Teil vor Ort am Lehrstuhl für Operations Research and Business Informatics durchgeführt. Teilaufgaben können nach Absprache von zu Hause erledigt werden. Der Student erhält einen eigenen Arbeitsplatz und ist aufgefordert sich am wissenschaftlichen Diskurs des Lehrstuhls zu beteiligen. Das Praktikum kann zu einem beliebigen Zeitpunkt begonnen werden (auch in den Semesterferien). Die Themenabsprache erfolgt mit dem jeweiligen Betreuer.

Weitere Informationen

Unterrichtssprache: deutsch / englisch

Literatur: Frederick Hillier, Gerald Lieberman, Introduction to Operations Research, edn.10, Mcgraw-Hill, 2014

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Tutortätigkeit					**CS 690**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
Ab 2.		jedes Semester	1 Semester	2	4

Modulverantwortliche/r	Studiendekan
Dozent/inn/en	Qualifizierte Studierende
Zuordnung zum Curriculum	ab dem 2. Studiensemester frei wählbar Pflicht für Studierende im Förderprogramm
Zulassungsvoraussetzungen	Die Tutoren werden vom Dozenten ausgewählt, Voraussetzung ist, dass der Tutor die Lehrveranstaltung mit sehr guter Note absolviert hat und didaktisches Interesse und didaktische Befähigung erkennen lässt.
Leistungskontrollen / Prüfungen	.
Lehrveranstaltungen / SWS	Übungen 2 SWS Leitung von Übungsruppen mit bis zu 20 Studierenden
Arbeitsaufwand	Ein Tutor unterstützt eine Lehrveranstaltung (typischerweise Grundvorlesung oder Stammvorlesung) über der Zeitraum eines Semesters. Das beinhaltet im Einzelnen: 0) Erlernen der fachdidaktischen Aspekte der jeweiligen Lehrveranstaltung (4h) 1) Moderieren einer wöchentlichen Übungsgruppe (je 90 min) mit etwa 20 Studenten 2) Korrigieren der wöchentlichen Tests, die in den ersten 15 Minuten der Übungsgruppe geschrieben werden. 3) Wöchentliche Beratungsstunden (je 90 Minuten) für die Hörer der Vorlesung 4) Teilnahme an der wöchentlichen Teambesprechung der Vorlesung, an der das gesamte Lehrpersonal teilnimmt (je 45 Minuten)

	5) Mitwirkung an der Erstellung der Musterlösungen für die wöchentlichen Übungsblätter (je 90 Minuten) 6) Beantwortung von Fragen zum Vorlesungsstoff und zum Übungsblatt auf der Mailingliste der Vorlesung (60 Minuten pro Woche) 7) Einarbeitung in den Vorlesungsstoff (2 Stunden pro Woche) 8) Erfinden neuer Übungsaufgaben (1 Stunde pro Woche) 9) Klausuraufsicht und Klausurkorrektur (Zwischenklausur, Endklausur, Nachklausur, je 12 Stunden)
Modulnote	unbenotet

Lernziele / Kompetenzen

Tutoren lernen, wie Lehrveranstaltungen organisiert werden und welche methodischen Ziele dabei verfolgt werden. Sie lernen, komplexe fachliche Inhalte sowohl in einer größeren Gruppe (Übungsgruppe) als auch in individuellen Beratungsgesprächen zu vermitteln.
Vor Beginn ihrer Tätigkeit, werden die Tutoren in einem oder mehreren Kolloquien in die wesentlichen fachdidaktischen Aspekte der jeweiligen Lehrveranstaltung eingeführt.
Sie lernen in ihrer Tätigkeit, sich an das unterschiedliche Vorwissen und die unterschiedlichen intellektuellen Fähigkeiten der betreuten Studierenden anzupassen. Sie werden ermutigt, komplexe fachliche Zusammenhänge einfach, prägnant und wirkungsvoll zu vermitteln. Gegebenfalls lernen Sie auch die Vermittlung fachlicher Inhalte auf Englisch

Inhalt
Siehe Arbeitsaufwand und Lernziele

Weitere Informationen

Unterrichtssprache: Deutsch

Literatur:
Dem Themengebiet entsprechende wissenschaftliche Artikel in enger Absprache mit dem Dozenten

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Modul Methodik und Didaktik für Tutoren					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
3			1 Semester		1

Modulverantwortliche/r Prof. Dr. Holger Hermanns

Dozent/inn/en Prof. Dr. Holger Hermanns

Zuordnung zum Curriculum 3. Semester

Zulassungsvoraussetzungen Programmierung 1, Mathematik für Informatiker 1

Leistungskontrollen / Prüfungen Lehrprobe

Lehrveranstaltungen / SWS

Arbeitsaufwand 30 h = 12 h Präsenz, 18 h Selbststudium

Modulnote unbenotet

Lernziele / Kompetenzen Die Studenten lernen, erworbenes Fachwissen im Bereich der Informatik selbständig und effektiv anderen Studenten zu vermitteln und dabei (fach-) didaktische Methoden anzuwenden.

Inhalt

Didaktische Grundlagen:
- geeignete Sozialformen (Frontalunterricht vs. Gruppenarbeit, mögliche Zwischenformen etc.)
- Lern- und Lehrtheorien der Erwachsenenbildung
- Fachdidaktik

Grundlagen der Leistungsbeurteilung:
- verschiedene Bezugsnormen (soziale, sachliche, fähigkeitsorientierte etc.)
- Gütekriterien von Prüfungen (Objektivität, Reliabilität, Validität)
- Häufige Beurteilungsfehler (z.B. Halo-Effekt, Primacy und Recency-Effekt)
- Praktische Anwendung: Erstellung und Bewertung von Minitests

Relevante Soft-Skills:
- Diversität von Gruppenpersönlichkeiten (Erkennen und entsprechendes Verhalten)
- Rhetorik, Körpersprache
- Anleitung und Hilfestellung

Rechtliches:
- Schweigepflicht, Datenschutz
- Grundsatz der Gleichbehandlung
- Transparenz von Bewertungskriterien

Weitere Informationen

Unterrichtssprache: deutsch

Medienformen: Tafelvortrag, interaktive Übungen, Tablet-PC

Fakultät für Mathematik und Informatik
Bachelor-Studiengang Informatik

Sprachkurse					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1-5		jedes Semester	1 Semester	2-4 & indiv.	3/6

Modulverantwortliche/r Dr. Peter Tischer, Leiter des Sprachenzentrums

Dozent/inn/en http://www.szsb.uni-saarland.de/mitarbeiter/

Zuordnung zum Curriculum
Bachelor Informatik
Master Informatik
Wahl

Zulassungsvoraussetzungen
Für Anfänger: keine
Französisch, Englisch, Spanisch:
Obligatorischer Einstufungstest
Fortgeschrittenenkurse:
Nachweise über belegte Kurse bzw. Gespräch mit dem Dozenten.

Leistungskontrollen / Prüfungen
Abschlussklausur und Anwesenheit beim Unterricht (mindestens 80%)

Lehrveranstaltungen / SWS
Seminar mit 2 - 4 SWS, eigenständiges Lernen mit monatlichen Treffen und 4wöchige Intensivkurse mit 4 h Unterricht täglich.
Gruppen von 6 – 40 Studierenden

Arbeitsaufwand
90 h = 30 h Seminar und 60 h Eigenstudium
180 h = 80 h Seminar und 100 h Eigenstudium

Modulnote

Lernziele / Kompetenzen

Auf entsprechendem Niveau:
- Leseverstehen
- Hörverstehen
- Sprechfertigkeit
- Grammatik
- Schreibtraining

Inhalt
Abhängig vom Kurs

Weitere Informationen

Unterrichtssprache: Deutsch und unterrichtete Sprache

Literatur: Kursabhängig

Medienform: Bücher, Beamer, Folien, Tafel, Sprachlabor, Video