

Fachrichtung Informatik
Fakultät 6 – Mathematik und Informatik
Universität des Saarlandes
Modulhandbuch
Bachelor-Studiengang Informatik
(08.10.2008)

Präambel: 1

Modulbeschreibungen: Grundlagen	2
Modulbeschreibung: Perspektiven der Informatik	2
Modulbeschreibungen: Mathematik für Informatiker 1-3	3
Modulbeschreibung: Programmierung 1	9
Modulbeschreibung: Programmierung 2	11
Modulbeschreibung: Systemarchitektur	13
Modulbeschreibung: Grundzüge von Algorithmen und Datenstrukturen	15
Modulbeschreibung: Informationssysteme	17
Modulbeschreibung: Theorie und Praxis Nebenläufiger Programmierung	19
Modulbeschreibung: Grundzüge der Theoretischen Informatik	22
Modulbeschreibung: Softwaredesignpraktikum	24
Modulbeschreibung: Hardwaredesignpraktikum	26
Modulbeschreibung: Proseminar	28
Modulbeschreibung: Seminar	30
Modulbeschreibung: Bachelor-Seminar	32
Modulbeschreibung: Bachelorarbeit	34
Modulbeschreibung: Tutortätigkeit	35
Modulbeschreibung: Sprachkurs IT-English	37
Modulbeschreibung: Sprachkurse	39
Soft Skills	41
Modulbeschreibung: Methodik und Didaktik für Tutoren	41

Präambel:

Allgemeine Regelungen für Grund- und Stammvorlesungen

Die folgenden Regelungen sollen im Normalfall gelten. Ausnahmen sind möglich und notwendig, wie beispielsweise bei den Blockvorlesungen im März und September.

- 1) Es werden zwei abschließende Prüfungsmöglichkeiten angeboten, eine zum Ende der Vorlesungszeit (Endterm) und eine zum Semesterende (Final). Diese zweifache Prüfungsmöglichkeit soll den Studierenden dabei helfen, ihre Arbeitsbelastung gleichmäßig über ein Semester zu verteilen.
- 2) Studierende, die am Endterm teilnehmen, können zusätzlich am Final teilnehmen. In diesem Fall zählt die bessere Leistung.
- 3) Die Zulassung zum Endterm/Final wird in der Regel über eine Mindestzahl von Punkten geregelt, die beispielsweise durch Tests, Abgabe von Übungsaufgaben oder Teilnahme an einem Midterm erzielt werden.
- 4) Final und Endterm können als Teilprüfungen verstanden werden, damit beispielsweise die unter (3) aufgeführten Leistungen in die Endnote eingehen können.
- 5) Die genauen Prüfungsmodalitäten werden zu Vorlesungsbeginn vollständig und verbindlich auf der Webseite der Vorlesung bekannt gegeben.

Modulbeschreibungen: Grundlagen

Modulbeschreibung: Perspektiven der Informatik

Studiengang:	Bachelor Studiengang Informatik
Modulbezeichnung:	Ringvorlesung Perspektiven der Informatik
ggf. Kürzel	CS 101
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Vorlesung: 2 SWS
Semester:	1. Semester / Wintersemester
Modulverantwortliche(r):	Studiendekan der Fakultät Mathematik und Informatik bzw. Studienbeauftragter der Informatik
Dozent(in):	Professoren der Fachrichtung
Sprache:	Deutsch
Zuordnung zum Curriculum	1. Semester, Pflicht
Lehrform / SWS:	Vorlesung 2 SWS
Arbeitsaufwand:	120 h = 30 h Präsenz- und 90 h Eigenstudium
Kreditpunkte:	2
Voraussetzungen:	keine
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.
Studien-Prüfungsleistungen:	Positive Bewertung von mindestens drei schriftlichen Zusammenfassungen verschiedener Vorträge
Medienformen:	Präsentationen mit Laptop und Beamer
Literatur:	Hintergrundmaterial zu den einzelnen Vorträgen

Modulbeschreibungen: Mathematik für Informatiker 1-3

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Mathematik für Informatiker 1
ggf. Kürzel	CS 110 / Mfl 1
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Vorlesung: 4 SWS Übung: 2 SWS
Semester:	1. Semester / Wintersemester
Modulverantwortliche(r):	Prof. Dr. Joachim Weickert
Dozent(in):	Prof. Dr. Joachim Weickert, Prof. Dr. Frank-Olaf Schreyer, Prof. Dr. Wolfram Decker
Sprache:	Deutsch
Zuordnung zum Curriculum	1. Semester, Pflicht
Lehrform / SWS:	Vorlesung: 4 SWS Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand:	270 h = 80 h Präsenz- und 190 h Eigenstudium
Kreditpunkte:	9
Voraussetzungen:	keine
Lernziele / Kompetenzen:	<ul style="list-style-type: none"> • 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:	<p>Die Zahlen geben die Gesamtzahl der Doppelstunden an.</p> <p>DISKRETE MATHEMATIK UND EINDIMENSIONALE ANALYSIS</p> <p>A. Grundlagen der diskreten Mathematik (8)</p> <ol style="list-style-type: none"> 1. Mengen (1) 2. Logik (1) 3. Beweisprinzipien, incl. vollst. Induktion (1)

	<p>4. Relationen (1) 5. Abbildungen (2) - injektiv, surjektiv, bijektiv - Mächtigkeit, Abzählbarkeit - Schubfachprinzip 6. Primzahlen und Teiler (1) 7. Modulare Arithmetik (1)</p> <p>B. Eindimensionale Analysis (22)</p> <p>B.1 Zahlen, Folgen und Reihen (8) 8. Axiomatik der reellen Zahlen, sup, inf (1) 9. Komplexe Zahlen (1) 10. Folgen (1 1/2) 11. Landau'sche Symbole (1/2) 12. Reihen: Konvergenzkriterien, absolute Kgz. (2) 13. Potenzreihen (1/2) 14. Zahlendarstellungen (1/2) 15. Binomialkoeffizienten und Binomialreihe (1)</p> <p>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)</p> <p>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)</p>
Studien- Prüfungsleistungen:	<ul style="list-style-type: none"> • Teilnahme an den Übungen und Bearbeitung der wöchentlichen Übungsaufgaben (50 Prozent der Übungspunkte werden zur Klausurteilnahme benötigt) • Bestehen der Abschlussklausur oder der Nachklausur • Die Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.
Medienformen:	primär Tafelvorlesung, z.T. ergänzt durch Overhead- und Laptoppräsentationen
Literatur:	<ul style="list-style-type: none"> • P. Hartmann: Mathematik für Informatiker. Vieweg, 2003. • M.P.H. Wolff, P. Hauck, W. Küchlin: Mathematik für Informatik und Bioinformatik. Springer, 2004.

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Mathematik für Informatiker 2
ggf. Kürzel	CS 210/ Mfl 2
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Vorlesung: 4 SWS Übung: 2 SWS
Semester:	2. Semester / Sommersemester
Modulverantwortliche(r):	Prof. Dr. Joachim Weickert
Dozent(in):	Prof. Dr. Joachim Weickert, Prof. Dr. Frank-Olaf Schreyer, Prof. Dr. Wolfram Decker
Sprache:	Deutsch
Zuordnung zum Curriculum	2. Semester, Pflicht
Lehrform / SWS:	Vorlesung: 4 SWS Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand:	270 h = 80 h Präsenz- und 190 h Eigenstudium
Kreditpunkte:	9
Voraussetzungen:	Mfl 1
Lernziele / Kompetenzen:	<ul style="list-style-type: none"> • 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:	<p>Die Zahlen geben die Gesamtzahl der Doppelstunden an.</p> <p>ALGEBRAISCHE STRUKTUREN UND LINEARE ALGEBRA</p> <p>C. ALGEBRAISCHE STRUKTUREN (5)</p> <p>29. Gruppen (2)</p> <p>30. Ringe und Körper (1)</p> <p>31. Polynomringe über allgemeinen Körpern (1/2)</p> <p>32. Boole'sche Algebren (1/2)</p>

	<p>D. LINEARE ALGEBRA (21)</p> <p>33. Vektorräume (2)</p> <ul style="list-style-type: none"> - Def., Bsp., - lineare Abb. - Unterraum, - Erzeugnis, lineare Abhängigkeit, Basis, Austauschatz <p>34. Lineare Abb. (Bild, Kern) (1)</p> <p>35. Matrixschreibweise für lineare Abbildungen (1 1/2)</p> <ul style="list-style-type: none"> - Interpretation als lineare Abbildungen - Multiplikation durch Hintereinanderausführung - Ringstruktur - Inverses <p>36. Rang einer Matrix (1/2)</p> <p>37. Gauss-Algorithmus für lineare Gleichungssysteme: (2)</p> <ul style="list-style-type: none"> - Gausselimination (1) - Lösungstheorie (1) <p>38. Iterative Verfahren für lineare Gleichungssysteme (1)</p> <p>39. Determinanten (1)</p> <p>40. Euklidische Vektorräume, Skalarprodukt (1)</p> <p>41. Funktionalanalytische Verallgemeinerungen (1)</p> <p>42. Orthogonalität (2)</p> <p>43. Fourierreihen (1)</p> <p>44. Orthogonale Matrizen (1)</p> <p>45. Eigenwerte und Eigenvektoren (1)</p> <p>46. Eigenwerte und Eigenvektoren symmetrischer Matrizen (1)</p> <p>47. Quadratische Formen und positiv definite Matrizen (1)</p> <p>48. Quadriken (1)</p> <p>50. Matrixnormen und Eigenwertabschätzungen (1)</p> <p>51. Numerische Berechnung von Eigenwerten und Eigenvektoren (1)</p>
Studien- Prüfungsleistungen:	<ul style="list-style-type: none"> • Teilnahme an den Übungen und Bearbeitung der wöchentlichen Übungsaufgaben (50 Prozent der Übungspunkte werden zur Klausurteilnahme benötigt) • Bestehen der Abschlussklausur oder der Nachklausur • Die Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.
Medienformen:	primär Tafelvorlesung, z.T. ergänzt durch Overhead- und Laptopräsentationen
Literatur:	<ul style="list-style-type: none"> • P. Hartmann: Mathematik für Informatiker. Vieweg, 2003. • M.P.H. Wolff, P. Hauck, W. Küchlin: Mathematik für Informatik und BioInformatik. Springer, 2004.

Studiengang:	Bachelor Informatik
Modulbezeichnung:	Mathematik für Informatiker 3
ggf. Kürzel	CS 310/ Mfl 3
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Vorlesung: 4 SWS Übung: 2 SWS
Semester:	3. Semester / Wintersemester
Modulverantwortliche(r):	Prof. Dr. Joachim Weickert
Dozent(in):	Prof. Dr. Joachim Weickert, Prof. Dr. Frank-Olaf Schreyer, Prof. Dr. Wolfram Decker
Sprache:	Deutsch
Zuordnung zum Curriculum	3. Semester, Pflicht
Lehrform / SWS:	Vorlesung: 4 SWS Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand:	270 h = 80 h Präsenz- und 190 h Eigenstudium
Kreditpunkte:	9
Voraussetzungen:	Mfl1 und Mfl 2
Lernziele / Kompetenzen:	<ul style="list-style-type: none"> • 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:	<p>Die Zahlen geben die Gesamtzahl der Doppelstunden an.</p> <p>STOCHASTIK, NUMERIK UND MEHRDIMENSIONALE ANALYSIS</p> <p>E. NUMERISCHE ERGÄNZUNGEN (3)</p> <p>52. Banachscher Fixpunktsatz (1) 53. Interpolation, incl. Splines (2)</p> <p>F. MEHRDIMENSIONALE ANALYSIS UND NUMERIK (11)</p>

	<p>54. Stetigkeit und Differentialoperatoren für skalarwertige Funktionen (2)</p> <p>55. Differentialoperatoren für vektorwertige Funktionen (1)</p> <p>56. Totale Differenzierbarkeit (1/2)</p> <p>57. Mittelwertsatz und Satz von Taylor (1 1/2)</p> <p>58. Extrema von Funktionen mehrerer Variabler (1)</p> <p>59. Das Newton-Verfahren (1)</p> <p>60. Extrema mit Nebenbedingungen (1)</p> <p>61. Mehrfachintegrale (1)</p> <p>62. Die Umkehrfunktion und die Transformationsregel (1)</p> <p>63. Variationsrechnung (1)</p> <p>G. STOCHASTIK (16)</p> <p>64. Grundbegriffe (Ws., Stichprobenraum) (1/3)</p> <p>65. Kombinatorik (2/3)</p> <p>66. Erzeugende Funktionen (1)</p> <p>67. Bedingte Wahrscheinlichkeiten (1)</p> <p>68. Zufallsvariable, Erwartungswert, Varianz (2) (Systemzuverlässigkeit, Varianz, Kovarianz, Jensen)</p> <p>69. Abschätzungen für Abweichungen vom Mittelwert (1) (Momente, Schranken von Markov, Chebyshev, Chernoff, schwaches Gesetz der grossen Zahlen)</p> <p>70. Wichtige diskrete Verteilungen (1)</p> <p>71. Wichtige kontinuierliche Verteilungen (1) (incl. zentraler Grenzwertsatz)</p> <p>72. Multivariate Verteilungen und Summen von Zufallsvariablen (1)</p> <p>73. Parameterschätzung und Konfidenzintervalle (1)</p> <p>74. Hypothesentests (1)</p> <p>75. Methode der kleinsten Quadrate (1)</p> <p>76. Robuste Statistik (2/3)</p> <p>77. Fehlerfortpflanzung (1/3)</p> <p>78. Markowketten (2)</p> <p>79. Pseudozufallszahlen und Monte-Carlo-Simulation (1)</p>
Studien- Prüfungsleistungen:	<ul style="list-style-type: none"> • Teilnahme an den Übungen und Bearbeitung der wöchentlichen Übungsaufgaben (50 Prozent der Übungspunkte werden zur Klausurteilnahme benötigt) • Bestehen der Abschlussklausur oder der Nachklausur • Die Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.
Medienformen:	primär Tafelvorlesung, z.T. ergänzt durch Overhead- und Laptoppräsentationen
Literatur:	<ul style="list-style-type: none"> • P. Hartmann: Mathematik für Informatiker. Vieweg, 2003. • M.P.H. Wolff, P. Hauck, W. Küchlin: Mathematik für Informatik und Bioinformatik. Springer, 2004.

Modulbeschreibung: Programmierung 1

Studiengang:	Bachelor Studiengang
Modulbezeichnung:	Programmierung 1
ggf. Kürzel	CS 120 / P1
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Vorlesung: 4 SWS Übung: 2 SWS
Semester:	1. Semester / Wintersemester
Modulverantwortliche(r):	Prof. Dr. Gert Smolka
Dozent(in):	Prof. Dr. Gert Smolka, Prof. Dr. Andreas Podelski, Prof. Dr.-Ing. Holger Hermanns
Sprache:	Deutsch
Zuordnung zum Curriculum	1. Semester, Pflicht
Lehrform / 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
Kreditpunkte:	9
Voraussetzungen:	keine
Lernziele / Kompetenzen:	<ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none"> • 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)
Studien- Prüfungsleistungen:	<ul style="list-style-type: none"> • zwei Klausuren (Mitte und Ende der Vorlesungszeit) • Die Note wird aus den Klausuren gemittelt und kann durch Leistungen in den Übungen verbessert werden. • Eine Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.
Medienformen:	Tafelvortrag, Papier (Script und Übungsblätter) , Übungen am Computer
Literatur:	Skript zur Vorlesung; siehe auch Literaturliste vom WS 02/03: http://www.ps.uni-sb.de/courses/prog-ws02/literatur.html

Modulbeschreibung: Programmierung 2

Studiengang:	Bachelor Studiengang
Modulbezeichnung:	Programmierung 2
ggf. Kürzel	CS 220 / P2
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Vorlesung: 2 SWS Übung: 4 SWS
Semester:	2. Semester / Sommersemester
Modulverantwortliche(r):	Prof. Dr. Andreas Zeller
Dozent(in):	Prof. Dr. Andreas Zeller und andere
Sprache:	Deutsch
Zuordnung zum Curriculum	2. Semester, Pflicht
Lehrform / SWS:	Vorlesung: 2 SWS Übung: 4 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand:	270 h = 45 h Präsenz- und 225 h Eigenstudium
Kreditpunkte:	9
Voraussetzungen:	Programmierung 1
Lernziele / Kompetenzen:	<p>Die Studierenden lernen die Grundprinzipien der imperativen /objektorientierten Programmierung kennen. Dabei wird primär Java als Programmiersprache verwendet.</p> <p>In dieser Vorlesung lernen sie:</p> <ul style="list-style-type: none"> • mittelgroße objektorientierte Systeme in Java zu implementieren und zu testen • kleinere, wohlstrukturierte Programme in C++ zu schreiben - im Wesentlichen als Umsetzung/Übersetzung der entsprechenden Java-Konzepte • sich in wenigen Tagen eine neue imperative/objektorientierte Sprache anzueignen, um sich in ein bestehendes Projekt einzuarbeiten

Inhalt:	<ul style="list-style-type: none"> • Objekte und Klassen • Klassendefinitionen • Objektinteraktion • Objektsammlungen • Objekte nutzen und testen • Vererbung • Dynamische Bindung • Fehlerbehandlung • Graphische Oberflächen • Klassendesign und Modularität • Objekte in C++ • Systemnahe Programmierung <p>sowie spezifische Vorlesungen für die Programmieraufgaben.</p>
Studien- Prüfungsleistungen:	<p>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.</p> <p>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.</p> <p>Im Vorlesungsteil müssen die Studierenden eine Klausur 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.</p> <p>Im Praktikumsteil kann eine Nachaufgabe angeboten werden; im Vorlesungsteil eine Nachprüfung. Hiermit können Studierende nachträglich die Veranstaltung bestehen.</p>
Medienformen:	<p>Vorlesung: Folien + Lehrbücher + Tafel Übungen: Programmieraufgaben am Computer, Übungsaufgaben auf Papier und in Gruppen an der Tafel</p>
Literatur:	<p>Java</p> <ul style="list-style-type: none"> • David J. Barnes & Michael Kölling: <i>Java lernen mit BlueJ</i> • Bruce Eckel: <i>Thinking in Java</i> • Joshua Bloch, <i>Effective Java</i> <p>C++</p> <ul style="list-style-type: none"> • Mark Allen Weiss: <i>C++ for Java programmers</i>

Modulbeschreibung: Systemarchitektur

Studiengang:	Bachelor Studiengang
Modulbezeichnung:	Systemarchitektur
ggf. Kürzel	CS 230 / SysArch
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Vorlesung: 4 SWS Übung: 2 SWS
Semester:	2. Semester / Sommersemester
Modulverantwortliche(r):	Prof. Dr. W.-J. Paul
Dozent(in):	Prof. Dr. W.-J. Paul
Sprache:	Deutsch
Zuordnung zum Curriculum	2. Semester, Pflicht
Lehrform / SWS:	Vorlesung: 4 SWS Übung: 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand:	270 h = 80 h Präsenz- und 190 h Eigenstudium
Kreditpunkte:	9
Voraussetzungen:	Empfohlen: Programmierung 1 und Mathematik für Informatiker 1
Lernziele / Kompetenzen:	Die Studierenden sollen lernen, ein einfaches System, bestehend aus Prozessor, Compiler und Betriebssystemkern <ul style="list-style-type: none"> • zu spezifizieren • zu designen und • per Korrektheitsbeweis (weil das am schnellsten geht) zu erklären, warum es funktioniert
Inhalt:	<ol style="list-style-type: none"> 1. Hardware <ol style="list-style-type: none"> a. Boole'sche Algebra und Schaltkreise b. Elementare Rechnerarithmetik c. ALU (Konstruktion und Korrektheit) d. Sequentieller vereinfachter DLX-Prozessor (Konstruktion und Korrektheit) 2. Compiler für eine C-ähnliche Sprache <ol style="list-style-type: none"> a. Syntax b. Semantik

	<ul style="list-style-type: none"> c. Korrektheitskriterium für die Übersetzung d. Code-Generierung e. Ausgewählte Teile des Korrektheitsbeweises <p>3. Betriebssystemkern</p> <ul style="list-style-type: none"> a. Interrupt-Mechanismus b. Memory-Management-Unit c. Spezifikation des Verhaltens (CVM) d. Konstruktion e. Ausgewählte Teile des Korrektheitsbeweises
Studien- Prüfungsleistungen:	<p>Studienleistungen: die Vorlesungen hören, nachbearbeiten und gegebenenfalls verstehen; die Übungen allein oder in Gruppen bearbeiten; erfolgreich bearbeitete Übungen in der Übungsgruppe vortragen und gegebenenfalls erklären können.</p> <p>Prüfungsleistungen: erfolgreiche Bearbeitung von 50 % der Übungsaufgaben berechtigt zur Teilnahme an den Klausuren (2 Klausuren, 1 Nachklausur)</p> <p>Die Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.</p>
Medienformen:	Tafelvortrag, gelegentlich unterstützt von Overhead-Folien und Beamer.
Literatur:	<p>Für den Hardware-Teil</p> <ul style="list-style-type: none"> • Keller-Paul: Hardware-Design; Teubner • Müller-Paul: Computer-Architecture; Springer <p>Für den Rest: Eigene wissenschaftliche Arbeiten. Mitschrift der Vorlesung auf der Web-Site der Vorlesung</p>

Modulbeschreibung: Grundzüge von Algorithmen und Datenstrukturen

Studiengang:	Bachelor Studiengang
Modulbezeichnung:	Grundzüge von Algorithmen und Datenstrukturen
ggf. Kürzel	CS 340 / GrADS
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Vorlesung: 2 SWS Übung: 2 SWS
Semester:	3. Semester / Wintersemester
Modulverantwortliche(r):	Prof. Dr. Raimund Seidel,
Dozent(in):	Prof. Dr. Markus Bläser, Prof. Dr. Kurt Mehlhorn, Prof. Dr. Raimund Seidel
Sprache:	Deutsch
Zuordnung zum Curriculum	3. Semester, Pflicht
Lehrform / SWS:	Vorlesung: 2 SWS Übung: 2 SWS in Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand:	180 h = 60 h Präsenz- und 120 h Eigenstudium
Kreditpunkte:	6
Voraussetzungen:	Programmierung 1 und 2 Mathematik für Informatiker 1 und 2
Lernziele / Kompetenzen:	<p>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.</p> <p>Die Studierenden lernen wichtige effiziente Datenstrukturen und Algorithmen kennen. Sie sollen die Fähigkeit erwerben, vorhandene Methoden durch theoretische Analysen und</p>

	<p>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.</p>
--	---

Modulbeschreibung: Informationssysteme

Studiengang:	Bachelor Studiengang
Modulbezeichnung:	Informationssysteme
ggf. Kürzel	CS 330 / InfoSys
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Vorlesung: 3 SWS Übung: 1 SWS
Semester:	4. Semester / Sommersemester
Modulverantwortliche(r):	Prof. Dr. Gerhard Weikum
Dozent(in):	Prof. Dr. Gerhard Weikum, Prof. Dr. Christoph Koch
Sprache:	Deutsch
Zuordnung zum Curriculum	4. Semester, Pflicht
Lehrform / SWS:	Vorlesung: 3 SWS Übung: 1 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand:	180 h = 80 h Präsenz- und 100 h Eigenstudium
Kreditpunkte:	6
Voraussetzungen:	Keine
Lernziele / Kompetenzen:	Die Vorlesung vermittelt grundlegende Kenntnisse über Konzepte und Schnittstellen von Datenbanksystemen und anderen Arten von Informationsdienstsoftware sowie der Anwendungsentwicklungswerkzeuge zur Realisierung von Informationssystemen. Besonderes Augenmerk wird auf die logische Ebene des ANSI 3-Schichtenmodells gelegt.
Inhalt:	Schwerpunktthemen sind das relationale Modell, Anfragesprachen für Datenbanksysteme, Nichtausdrückbarkeitsbeweise, Datenmodellierung, Designtheorie und Normalformen fuer relationale Schemata, Aequivalenz und Minimierung von Anfragen, Integritaetsbedingungen, Datenintegration und aktuelle Themen wie Webinformationssysteme, Information Retrieval, und die Handhabung von unvollstaendiger Information.

	Die notwendigen Grundlagen werden in der Vorlesung eingeführt.
Studien- Prüfungsleistungen:	<p>Es werden 6 benotete Leistungspunkte vergeben, wenn folgende Voraussetzungen erfüllt sind:</p> <ol style="list-style-type: none"> 1. erfolgreiche Teilnahme an zwei Teilklausuren in der Mitte und am Ende des Semesters oder erfolgreiche Teilnahme an einer Teilklausur und der Nachklausur Anfang Oktober 2. erfolgreiche Teilnahme an den Übungen: Abgabe eines kleinen Programmierprojekts und Erreichen von mehr als der Hälfte der möglichen Punkte bei kurzen Multiple-Choice-Tests in den Übungsstunden. <p>Die Note wird aus den Ergebnissen der zwei bestandenen (Teil-) Klausuren berechnet.</p>
Medienformen:	Tafelvortrag und Präsentationen mit Laptop / Beamer
Literatur:	<ul style="list-style-type: none"> • Alfons Kemper, Andre Eickler: Datenbanksysteme - eine Einführung, Oldenbourg, 2001 • Serge Abiteboul, Richard Hull, Victor Vianu: Foundations of Databases, Addison-Wesley, 1995 • Jiawei Han, Micheline Kamber: Data Mining - Concepts and Techniques, Morgan Kaufmann, 2001

Modulbeschreibung: Theorie und Praxis Nebenläufiger Programmierung

Studiengang:	Bachelor Studiengang Informatik
Modulbezeichnung:	Theorie und Praxis Nebenläufiger Programmierung
ggf. Kürzel	
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Vorlesung 2 SWS Übung 2 SWS
Semester:	4 . Semester
Modulverantwortliche(r):	Prof. Dr. Ing. Holger Hermanns
Dozent(in):	Prof. Dr.-Ing. Holger Hermanns
Sprache:	Deutsch
Zuordnung zum Curriculum	3.-5. Semester Bachelor CS, oder 1.-3. Semester Master CS
Lehrform / SWS:	Vorlesung: 2 SWS (ca. 25 Studierende) Übung: 2 SWS (Übungsgruppe mit bis zu 25 Studierenden)
Arbeitsaufwand:	180 h = 80 h Präsenz, 100 h Selbststudium
Kreditpunkte:	6
Voraussetzungen:	Programmierung 1 & 2, Softwarepraktikum
Lernziele / Kompetenzen:	Dies ist eine einmalige Spezialvorlesung, die dazu dient, die Basisvorlesung „Nebenläufige Programmierung“ inhaltlich vorzubereiten.
Inhalt:	<p>Nebenläufigkeit als Konzept</p> <ul style="list-style-type: none"> -- Potentieller Parallelismus -- Tatsächlicher Parallelismus -- Konzeptioneller Parallelismus <p>Nebenläufigkeit in der Praxis</p> <ul style="list-style-type: none"> -- Objektorientierung -- Betriebssysteme -- Multi-core Prozessoren, Coprozessoren -- Programmierte Parallelität -- Verteilte Systeme (client-server, peer-2-peer, Datenbanken, Internet)

	<p>Die Schwierigkeit von Nebenläufigkeit</p> <ul style="list-style-type: none"> -- Ressourcenkonflikte -- Fairness -- Gegenseitiger Ausschluss -- Verklemmung (Deadlock) -- gegenseitige Blockaden (Livelock) -- Verhungern (Starvation) <p>Grundlagen der Nebenläufigkeit</p> <ul style="list-style-type: none"> -- Sequentielle Prozesse -- Zustände, Ereignisse und Transitionen -- Transitionssysteme -- Beobachtbares Verhalten -- Determinismus vs. Nicht-Determinismus -- Algebren und Operatoren <p>CCS: Der Kalkül kommunizierender Prozesse</p> <ul style="list-style-type: none"> -- Konstruktion von Prozessen: Sequenz, Auswahl, <p>Rekursion</p> <ul style="list-style-type: none"> -- Nebenläufigkeit -- Interaktion -- Strukturelle operationelle Semantik -- Gleichheit von Beobachtungen -- Implementierungsrelationen -- CCS mit Datentransfer <p>Programmieren von Nebenläufigkeit</p> <ul style="list-style-type: none"> -- Java vs. C++ -- Objekte in Java -- Sockets, Protokolle, Datenströme in Java -- Shared Objects und Threads in Java -- Shared Objects und Threads als Transitionssysteme -- Monitore und Semaphoren <p>Analyse und Programmierunterstützung</p> <ul style="list-style-type: none"> -- Erkennung von Verklemmungen -- Zusicherung von Sicherheit und Lebendigkeit -- Model-Basiertes Design von Nebenläufigkeit -- Software Architekturen für Nebenläufigkeit
Studien- Prüfungsleistungen:	Schriftliche Ausarbeitung über ein Thema der Vorlesung. Klausur am Ende der Vorlesung. In der Regel wird eine Nachprüfung angeboten (schriftlich oder mündlich).
Medienformen:	Tafelvortrag, Tablet-PC, Übungsblätter, Skript, Übungen am Computer
Literatur:	<p>Jeff Magee and Jeff Kramer. Concurrency: State Models & Java Programs (2nd Edition), Wiley, 2006 http://www-dse.doc.ic.ac.uk/concurrency/</p>

	<p>Robin Milner. Communication and Concurrency , Prentice Hall, 1995</p> <p>Luca Aceto, Anna Ingolfsdottir, Kim Guldstrand Larsen, and Jiri Srba. An introduction to Milner's CCS, 2005. http://www.cs.auc.dk/~luca/SV/intro2ccs.pdf</p> <p>Howard Bowman and Rodolfo Gomez, Concurrency theory: Calculi and automata for modelling untimed & timed concurrent systems. Springer, 2006.</p>
--	---

Modulbeschreibung: Grundzüge der Theoretischen Informatik

Studiengang:	Bachelor Studiengang Informatik
Modulbezeichnung:	Grundzüge der Theoretischen Informatik
ggf. Kürzel	CS 420 / Theolnf
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Vorlesung: 4 SWS Übung: 2 SWS
Semester:	3. Semester / Wintersemester
Modulverantwortliche(r):	Prof. Dr. Raimund Seidel
Dozent(in):	Prof. Dr. Bernd Finkbeiner, Prof. Dr. Kurt Mehlhorn, Prof. Dr. W.J. Paul, Prof. Dr. Raimund Seidel, Prof. Dr. Reinhard Wilhelm
Sprache:	Deutsch
Zuordnung zum Curriculum	3. Semester, Pflicht
Lehrform / SWS:	Vorlesung 4 SWS Übung 2 SWS Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand:	270 h = 80 h Präsenz- und 190 h Eigenstudium
Kreditpunkte:	9
Voraussetzungen:	Programmierung 1, Programmierung 2, Mfl 1-2
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;

	<p>Turing Maschinen und äquivalente Modelle von allgemeiner Berechenbarkeit (z.B. μ-rekursive Funktionen, Random Access Machines)</p> <p>Reduzierbarkeit, Entscheidbarkeit, Nicht-Entscheidbarkeit;</p> <p>Die Komplexitätsmaße Zeit und Platz; die Komplexitätsklassen P und NP; Grundzüge der Theorie der NP-Vollständigkeit</p>
Studien- Prüfungsleistungen:	<ul style="list-style-type: none"> • Erfolgreiche Bearbeitung der Übungsaufgaben; • Bestehen von Zwischentests und einer Abschlussklausur • Benotung: Ja. • Die Note ergibt sich aus den Tests und dem Klausurergebnis. • Eine Nachklausur findet innerhalb der letzten beiden Wochen vor Vorlesungsbeginn des Folgesemesters statt.
Medienformen:	Tafelvortrag und Präsentationen mit Laptop / Beamer
Literatur:	<ul style="list-style-type: none"> • Ingo Wegener: Theoretische Informatik - eine algorithmenorientierte Einführung. • Harry R. Lewis, Christos H. Papadimitriou: Elements of the Theory of Computation • John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman: Introduction to Automata Theory, Languages, and Computation • Uwe Schöning: Theoretische Informatik - kurzgefasst • Michael Sipser: Introduction to the Theory of Computation • Norbert Blum: Theoretische Informatik

Modulbeschreibung: Softwaredesignpraktikum

Studiengang:	Bachelor Studiengang Informatik
Modulbezeichnung:	Softwarepraktikum
ggf. Kürzel	CS 320 / SoDePra
ggf. Untertitel	Praktikum 9 LP
ggf. Lehrveranstaltungen:	3. Semester (Vorlesungsfreie Zeit nach 2. Semester)
Semester:	Prof. Dr. Andreas Zeller
Modulverantwortliche(r):	Prof. Dr. Andreas Zeller, Prof. Dr. Philipp Slusallek, Prof. Dr. Holger Hermanns
Dozent(in):	Deutsch
Sprache:	3.Semester (Sommersemesterferien), Pflicht <ul style="list-style-type: none"> • Angewandte Informatik (Diplom): 2. Semester Pflicht (läuft aus) • Bioinformatik (BSc): 2. Semester, Pflicht • Computer- und Kommunikationstechnik (Diplom): 4. Semester, Pflicht • Informatik (Diplom): 2.Semester, Pflicht (läuft aus)
Zuordnung zum Curriculum	Vorlesung 2 SWS Praktikum 4 SWS (Teams in Gruppen bis zu 6 Studierende)
Lehrform / SWS:	270 h = 20 h Präsenz- und 250 h Eigenstudium
Arbeitsaufwand:	9
Kreditpunkte:	Bachelor Studiengang Informatik
Voraussetzungen:	Programmieren 1 und 2
Lernziele / Kompetenzen:	<p>Die Studierenden erwerben die Fähigkeit, im Team zu arbeiten und Probleme der Informatik zu lösen.</p> <p>Die Studierenden wissen, welche Probleme beim Durchführen eines Software-Projekts auftreten können, und wie man damit umgeht.</p> <p>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.</p>

	<p>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.</p> <p>Sie beherrschen Techniken der Qualitätssicherung wie Testen und Gegenlesen und wenden diese an.</p>
Inhalt:	<p>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)</p>
Studien- Prüfungsleistungen:	<ul style="list-style-type: none"> • Erfolgreiches Erstellen im Team eines komplexen Software-Produkts, insbesondere <ul style="list-style-type: none"> ○ Einreichen der erforderlichen Dokumente ○ Abnahme des Endprodukts durch den Kunden ○ Einhaltung der Termin- und Qualitätsstandards
Medienformen:	<p>Gruppenarbeit am Rechner Präsentation mit Tafel und Folie Demonstration für den Kunden</p>
Literatur:	<p>Balzert, Einführung in die Softwaretechnik I + II Gamma et al., Entwurfsmuster</p>

Modulbeschreibung: Hardwaredesignpraktikum

Studiengang:	Bachelor -Studiengang Informatik
Modulbezeichnung:	Hardwaredesignpraktikum
ggf. Kürzel	CS 321 / HaDePra
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Praktikum: 6 SWS
Semester:	2.-4. Semester
Modulverantwortliche(r):	Prof. Dr. W.-J. Paul
Dozent(in):	Jan Dörrenbächer
Sprache:	Deutsch
Zuordnung zum Curriculum	2.-4. Semester, Wahl
Lehrform / SWS:	Praktikum 6 SWS
Arbeitsaufwand:	180 h = 90 h Präsenz- und 90 h Selbststudium
Kreditpunkte:	6
Voraussetzungen:	Empfohlen: Systemarchitektur
Lernziele / Kompetenzen:	<ul style="list-style-type: none"> - Das praktische Designen und Debuggen einfacher Prozessoren und Platinen
Inhalt:	<ul style="list-style-type: none"> - Datenblätter, CAD-System - FPGA-Programmierung - Sequentielles DLX-Prozessordesign eingeben (ist bekannt aus Systemarchitektur-Vorlesung) - Entwicklung von Testsoftware; Debuggen des Designs - DLX-Assemblerroutinen
Studien-Prüfungsleistungen:	<p>Studienleistungen: die Termine des Praktikums besuchen, die Versuche in Gruppen durchführen, nachbearbeiten, die nächsten Versuche vorbereiten. Erfolgreiche Durchführung der Versuche berechtigt zur Teilnahme an Abschlussklausur und Nachklausur. Teilweise mündliche Prüfungen.</p>

Medienformen:	Laborpraktikum
Literatur:	<ul style="list-style-type: none">- Keller-Paul: Harwaredesign- Müller-Paul: Computer Architecture: Correctness and Complexity; Springer 2000

Modulbeschreibung: Proseminar

Studiengang:	Bachelor Studiengang Informatik
Modulbezeichnung:	Proseminar
ggf. Kürzel	CS 300
ggf. Untertitel	Wechselnde Titel je nach Thema
Semester:	4. Semester /Jedes Semester
Modulverantwortliche(r):	Studiendekan der Fakultät Mathematik und Informatik bzw. Studienbeauftragter der Informatik
Dozent(in):	Professoren der Fachrichtung
Sprache:	Deutsch
Zuordnung zum Curriculum	4. Semester (laut Beispielstudienplan, Angebot jedes Semester), Pflicht
Lehrform / SWS:	Proseminar 2 SWS (bis zu 25 Studierende)
Arbeitsaufwand:	150h = 40 h Präsenz und 110 h Eigenstudium
Kreditpunkte:	5
Voraussetzungen:	Keine
Lernziele / Kompetenzen:	<p>Die Studierenden haben am Ende der Veranstaltung ein profundes Verständnis aktueller oder fundamentaler Aspekte eines spezifischen Teilbereiches der Informatik erlangt.</p> <p>Sie haben Kompetenz im Verstehen einfacher wissenschaftlicher Aufsätze und im Präsentieren von wissenschaftlichen Erkenntnissen erworben.</p>
Inhalt:	<p>Praktisches Einüben unter Anleitung von</p> <ul style="list-style-type: none"> • Lesen und Verstehen wissenschaftlicher Aufsätze, • Diskutieren der Aufsätze in der Gruppe, • Analysieren, Zusammenfassen und Wiedergeben des spezifischen Themas, • Präsentationstechnik, <p>Spezifische Vertiefung in Bezug auf das individuelle Thema des Seminars.</p>
Studien-Prüfungsleistungen:	<ul style="list-style-type: none"> • Diskussion in der Gruppe • thematischer Vortrag

	<ul style="list-style-type: none">• kurze schriftliche Ausarbeitung
Medienformen:	typischerweise Folien- oder Tafelvorträge
Literatur:	dem Thema entsprechend

Modulbeschreibung: Seminar

Studiengang:	Bachelor Studiengang Studiengang Informatik
Modulbezeichnung:	Seminar
ggf. Kürzel	CS 500
ggf. Untertitel	Wechselnde Titel je nach Thema
ggf. Lehrveranstaltungen:	Seminar: 3 SWS
Semester:	5. Semester / Jedes Semester
Modulverantwortliche(r):	Studiendekan der Fakultät Mathematik und Informatik bzw. Studienbeauftragter der Informatik
Dozent(in):	Professoren der Fachrichtung
Sprache:	Deutsch / Englisch
Zuordnung zum Curriculum	5. Semester (laut Beispielstudienplan, Angebot jedes Semester), Pflicht
Lehrform / SWS:	Seminar 3 SWS (bis zu 25 Studierende)
Arbeitsaufwand:	210 h = 60 h Präsenz und 150 h Eigenstudium
Kreditpunkte:	7
Voraussetzungen:	Grundlegende Kenntnisse im jeweiligen Teilbereich der Informatik.
Lernziele / Kompetenzen:	<p>Die Studierenden haben am Ende der Veranstaltung ein tiefes Verständnis aktueller oder fundamentaler Aspekte eines spezifischen Teilbereiches der Informatik erlangt.</p> <p>Sie haben Kompetenz im eigenständigen wissenschaftlichen Recherchieren, Einordnen, Zusammenfassen, Diskutieren, Kritisieren und Präsentieren von wissenschaftlichen Erkenntnissen gewonnen.</p>
Inhalt:	<p>Praktisches Einüben von</p> <ul style="list-style-type: none"> • reflektierender wissenschaftlicher Arbeit, • Analyse und Bewertung wissenschaftlicher Aufsätze, • Verfassen eigener wissenschaftlicher Zusammenfassungen • Diskussion der Arbeiten in der Gruppe • Erarbeiten gemeinsamer Standards für wissenschaftliche Arbeiten

	<ul style="list-style-type: none"> • Präsentationstechnik <p>Spezifische Vertiefung in Bezug auf das individuelle Thema des Seminars.</p> <p>Der typische Ablauf eines Seminars ist wie folgt:</p> <ul style="list-style-type: none"> • Vorbereitende Gespräche zur Themenauswahl • Regelmäßige Treffen mit Diskussion ausgewählter Beiträge • Vortrag und Ausarbeitung zu einem der Beiträge • Mündliche Prüfung über das erarbeitete Themengebiet
Studien- Prüfungsleistungen:	<ul style="list-style-type: none"> • Beiträge zur Diskussion • Thematischer Vortrag • Schriftliche Ausarbeitung • Mündliche Abschlussprüfung über das gesamte Themengebiet
Medienformen:	<p>Diskussion in der Gruppe Typischerweise Folien- oder Tafelvorträge</p>
Literatur:	dem Thema entsprechend

Modulbeschreibung: Bachelor-Seminar

Studiengang:	Bachelor Studiengang Informatik
Modulbezeichnung:	Bachelor-Seminar
ggf. Kürzel	CS 690
ggf. Untertitel	
ggf. Lehrveranstaltungen:	Seminar 3 SWS Praktikum 2 SWS
Semester:	6. Semester / Jedes Semester
Modulverantwortliche(r):	Studiendekan der Fakultät Mathematik und Informatik bzw. Studienbeauftragter der Informatik
Dozent(in):	Professoren der Fachrichtung
Sprache:	Deutsch
Zuordnung zum Curriculum	6. Semester, Pflicht
Lehrform / 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
Kreditpunkte:	9
Voraussetzungen:	Gesamter Pflichtkanon des Bachelorstudiengangs, bis auf Bachelorseminar und –arbeit.
Lernziele / Kompetenzen:	<p>Im Bachelorseminar erwirbt der Studierende unter Anleitung die Fähigkeit zum wissenschaftlichen Arbeiten im Kontext eines angemessenen Themengebietes.</p> <p>Am Ende des Bachelorseminars sind die Grundlagen für eine erfolgreiche Anfertigung der Bachelorarbeit gelegt, und wesentliche Lösungsansätze bereits eruiert.</p> <p>Das Bachelorseminar bereitet somit die Themenstellung und Ausführung der Bachelorarbeit vor.</p> <p>Es vermittelt darüber hinaus praktische Fähigkeiten des wissenschaftlichen Diskurses. Diese Fähigkeiten werden durch die aktive Teilnahme an einem Lesekreis vermittelt, in welchem</p>

	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.
Studien- Prüfungsleistungen:	<ul style="list-style-type: none"> • 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.
Medienformen:	Divers
Literatur:	Dem Themengebiet entsprechende wissenschaftliche Artikel in enger Absprache mit dem Dozenten

Modulbeschreibung: Bachelorarbeit

Studiengang:	Bachelor Studiengang Informatik
Modulbezeichnung:	Bachelorarbeit
ggf. Kürzel	CS 699
ggf. Untertitel	
ggf. Lehrveranstaltungen:	
Semester:	6. Semester / Jedes Semester
Modulverantwortliche(r):	Studiendekan der Fakultät Mathematik und Informatik bzw. Studienbeauftragter der Informatik
Dozent(in):	Professoren der Fachrichtung
Sprache:	Deutsch / auf Antrag Englisch
Zuordnung zum Curriculum	6. Semester, Pflicht
Lehrform / SWS:	
Arbeitsaufwand:	360 h = 20 h Präsenz- und 340 h Eigenstudium
Kreditpunkte:	12
Voraussetzungen:	Erfolgreicher Abschluss des Bachelor-Seminars
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.
Studien-Prüfungsleistungen:	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.
Medienformen:	
Literatur:	Je nach Thema in Absprache mit dem Professor

Modulbeschreibung: Tutortätigkeit

Studiengang:	Bachelor / Master Studiengang Informatik
Modulbezeichnung:	Tutortätigkeit
ggf. Kürzel	CS-T
ggf. Untertitel	
ggf. Lehrveranstaltungen:	
Semester:	ab dem 2. Studiensemester / Jedes Semester
Modulverantwortliche(r):	Prof. Dr. Gert Smolka
Dozent(in):	Qualifizierte Studierende
Sprache:	Deutsch / English
Zuordnung zum Curriculum	ab dem 2. Studiensemester frei wählbar Pflicht für Studierende im Förderprogramm
Lehrform / SWS:	Übungen 2 SWS Leitung von Übungsgruppen mit bis zu 20 Studierenden
Arbeitsaufwand:	<p>Ein Tutor unterstützt eine Lehrveranstaltung (typischerweise Grundvorlesung oder Stammvorlesung) über der Zeitraum eines Semesters. Das beinhaltet im Einzelnen:</p> <ol style="list-style-type: none"> 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 der Vorlesungsstoff (2 Stunden pro Woche) 8) Erfinden neuer Übungsaufgaben (1 Stunde pro Woche) 9) Klausuraufsicht und Klausurkorrektur (Zwischenklausur, Endklausur, Nachklausur, je 12 Stunden)

Kreditpunkte:	4
Voraussetzungen:	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.
Lernziele / Kompetenzen:	<p>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.</p> <p>Vor Beginn ihrer Tätigkeit, werden die Tutoren in einem oder mehreren Kolloquien in die wesentlichen fachdidaktischen Aspekte der jeweiligen Lehrveranstaltung eingeführt.</p> <p>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.</p>
Inhalt:	Siehe Arbeitsaufwand und Lernziele
Studien- Prüfungsleistungen:	Der Dozent beobachtet die Tutoren bei ihren Beiträgen zu den Übungsaufgaben (neue erfinden, Musterlösungen für bestehende erstellen), bei der Beantwortung fachlicher Fragen auf der Mailingliste sowie der Klausurkorrektur und gibt ihnen Feedback. Der Assistent der Vorlesung besucht jede Übungsgruppe einmal im Semester, gibt den Tutoren Feedback und informiert den Dozenten. Die Studierenden evaluieren ihre Tutoren im Rahmen der Vorlesungsevaluation.
Medienformen:	Papier und Tafel
Literatur:	

Modulbeschreibung: Sprachkurs IT-English

Studiengang:	Bachelor Studiengang Informatik
Modulbezeichnung:	IT-English
ggf. Kürzel	
ggf. Untertitel	SZENXXXX
ggf. Lehrveranstaltungen:	Seminar 2 SWS
Semester:	1.-6. Semester
Modulverantwortliche(r):	Dr. Peter Tischer
Dozent(in):	Dr. Peter Tischer
Sprache:	English
Zuordnung zum Curriculum	Bachelor 1.- 5. Semester / Unicert III Wahlfach
Lehrform / SWS:	2 SWS Seminar, Gruppen bis zu 18 Studierende
Arbeitsaufwand:	90 h = 30 h Kurs und 60 h Eigenstudium
Kreditpunkte:	3
Voraussetzungen:	Studierende müssen einen Einstufungstest bestehen bzw. den erfolgreichen Besuch eines dem Unicert Niveaus III oder IV gleichwertigen Kurses nachweisen.
Lernziele / Kompetenzen:	Der Kurs soll Studierenden bei der Erweiterung ihres wissenschaftlichen und technischen Wortschatzes helfen. Hierbei werden spezielle Bedürfnisse und Interessen beachtet.
Inhalt:	Der Kurs beschäftigt sich mit Fach-English in den Bereichen IT und Ingenieurwissenschaften. Dabei ist nicht vorgesehen, systematisch Vokabellisten abzuarbeiten. Es werden vielmehr typische Texttypen wie z.B. Vorlesungen, Tabellen, Anleitungen, wissenschaftliche Aufsätze und Texte aus Lehrwerken betrachtet. Dabei werden auch für „Tech Talk“ typische grammatische Phänomene wie z.B. Passiv, Nominalisierung oder Konditionalsätze besprochen.
Studien-Prüfungsleistungen:	Regelmäßige Teilnahme (mindestens 80% aller Veranstaltungen) Ein Referat

	Abschlussklausur
Medienformen:	Siehe Inhalte Handouts, videos, etc.
Literatur:	Aktuelle Liste und Zusammenstellung von Material zu Beginn des Kurses

Modulbeschreibung: Sprachkurse

Studiengang:	Bachelor Studiengang Informatik
Modulbezeichnung:	Sprachkurse am Sprachenzentrum werden für die folgenden Sprachen angeboten: Chinesisch, Dänisch, Englisch, Französisch, Griechisch, Italienisch, Japanisch, Catalan, Koreanisch, Croatisch, Niederländisch, Polnisch, Portugiesisch, Russisch, Schwedisch und Spanisch.
ggf. Kürzel	
ggf. Untertitel	Hängt vom jeweiligen Kurs ab
ggf. Lehrveranstaltungen:	2-4 Stunden Seminar, aber auch Kurse mit betreutem Lernen oder unabhängigem Lernen
Semester:	1.-5. Semester
Modulverantwortliche(r):	Dr. Peter Tischer, Leiter des Sprachenzentrums
Dozent(in):	http://www.szb.uni-saarland.de/mitarbeiter/
Sprache:	Deutsch und unterrichtete Sprache
Zuordnung zum Curriculum	Wahl
Lehrform / 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
Kreditpunkte:	3 für 2 SWS 6 für 4 SWS 6 für einen Intensivkurs
Voraussetzungen:	Für Anfänger: keine Französisch, Englisch, Spanisch: Obligatorischer Einstufungstest Fortgeschrittenenkurse: Nachweise über belegte Kurse bzw. Gespräch mit dem Dozenten.
Lernziele / Kompetenzen:	Auf entsprechendem Niveau: <ul style="list-style-type: none"> • Leseverstehen • Hörverstehen • Sprechfertigkeit • Grammatik

	<ul style="list-style-type: none"> • Schreibtraining
Inhalt:	Abhängig vom Kurs
Studien- Prüfungsleistungen:	Abschlussklausur und Anwesenheit beim Unterricht (mindestens 80%)
Medienformen:	Bücher, Beamer, Folien, Tafel, Sprachlabor, Video
Literatur:	Kursabhängig

Soft Skills

Modulbeschreibung: Methodik und Didaktik für Tutoren

Studiengang:	Bachelor Studiengang Informatik
Modulbezeichnung:	Methodik und Didaktik für Tutoren
ggf. Kürzel	
ggf. Untertitel	
ggf. Lehrveranstaltungen:	
Semester:	3. Semester
Modulverantwortliche(r):	Prof. Dr. Holger Hermanns
Dozent(in):	Prof. Dr. Holger Hermanns
Sprache:	Deutsch
Zuordnung zum Curriculum	3. Semester
Lehrform / SWS:	
Arbeitsaufwand:	30 h = 12 h Präsenz, 18 h Selbststudium
Kreditpunkte:	1
Voraussetzungen:	Programmierung 1, Mathematik für Informatiker 1
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:	<p>Didaktische Grundlagen:</p> <ul style="list-style-type: none">- geeignete Sozialformen (Frontalunterricht vs. Gruppenarbeit, mögliche Zwischenformen etc.)- Lern- und Lehrtheorien der Erwachsenenbildung- Fachdidaktik <p>Grundlagen der Leistungsbeurteilung:</p> <ul style="list-style-type: none">- 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

	<p>Minitests</p> <p>Relevante Soft-Skills:</p> <ul style="list-style-type: none"> - Diversität von Gruppenpersönlichkeiten (Erkennen und entsprechendes Verhalten) - Rhetorik, Körpersprache - Anleitung und Hilfestellung <p>Rechtliches:</p> <ul style="list-style-type: none"> - Schweigepflicht, Datenschutz - Grundsatz der Gleichbehandlung - Transparenz von Bewertungskriterien
Studien- Prüfungsleistungen:	Lehrprobe
Medienformen:	Tafelvortrag, interaktive Übungen, Tablet-PC
Literatur:	