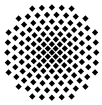

Universität Stuttgart

Modulhandbuch

Studiengang Bachelor of Science Informatik

Prüfungsordnung: 2009

Universität Stuttgart
Keplerstr. 7
70174 Stuttgart

Inhaltsverzeichnis

Präambel 4

100 Basismodule 5

12060 Datenstrukturen und Algorithmen 6
10190 Mathematik für Informatiker und Softwaretechniker 8
10260 Programmierkurs 10
10280 Programmierung und Software-Entwicklung 12
10930 Technische Grundlagen der Informatik 14
10940 Theoretische Grundlagen der Informatik 16

200 Kernmodule 18

10020 Algorithmik 19
14910 Berechenbarkeit und Komplexität 21
25610 Grundlagen des Software Engineerings 23
10210 Mensch-Computer-Interaktion 24
10220 Modellierung 26
10240 Numerische und Stochastische Grundlagen 28
10270 Programmierparadigmen 30
10310 Rechnerorganisation 32
10330 Systemkonzepte und -programmierung 34

300 Ergänzungsmodule 36

320 Katalog ISG 1-3 37
10140 Advanced Processor Architecture 38
10030 Architektur von Anwendungssystemen 40
10080 Datenbanken und Informationssysteme 42
10090 Grundlagen der Eingebetteten Systeme 44
10100 Grundlagen der Graphischen Ingenieursysteme 45
10110 Grundlagen der Künstlichen Intelligenz 47
10120 Grundlagen der Modellbildung und Simulation 49
10150 Grundlagen des Compilerbaus und der Programmiersprachen 51
10170 Imaging Science 53
10250 Parallele Systeme 55
11110 Verteilte Systeme 56
330 Katalog ISW 1-3 58
10050 Bildverstehen 59
10060 Computergraphik 61
11900 Design and Test of Systems on a Chip 63
10170 Imaging Science 65
10180 Information Retrieval und Text Mining 67
10250 Parallele Systeme 68
10300 Rechnernetze 69
340 Katalog ISW 4-7 71
10160 Hardwarebeschreibungssprachen 72
350 Wahlmodule aus Master Informatik 74
29560 Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems 75
10040 Bildsynthese 77
29570 Computer Interface Technologien 79
29580 Datenkompression 80
29590 Digitaler Systementwurf I 81
29440 Geometrische Modellierung und Animation 82
29610 Hardware Based Fault Tolerance 84

14380 Hardware Verification and Quality Assessment 86
29640 Mikrocontroller 88
29500 Visual Computing 90
11330 Visualisierung 92

400 Schlüsselqualifikationen fachaffin 94

10290 Projekt-INF 95
10320 Seminar-INF 97

700 Zusatzmodule 99

Präambel

Die Studien- und Prüfungsordnung für den Bachelorstudiengang Informatik beschreibt den Aufbau des Studiums und die Organisation der Prüfungen. Sie stellt das Regelwerk und die Rechtsgrundlage für eine einheitliche Handhabung des Studienablaufs und der Bewertung der Studien- und Prüfungsleistungen dar. Sie wendet sich dabei sowohl an die Studierenden als auch an die Prüfenden sowie an die entsprechenden Organe der Universität Stuttgart.
Aus Gründen der sprachlichen Vereinfachung wird grundsätzlich nur die grammatikalisch männliche Form für Personen verwendet. Sinngemäß ist stets auch die entsprechende weibliche Form gemeint.

100 Basismodule

Zugeordnete Module:

12060	Datenstrukturen und Algorithmen
10190	Mathematik für Informatiker und Softwaretechniker
10260	Programmierkurs
10280	Programmierung und Software-Entwicklung
10930	Technische Grundlagen der Informatik
10940	Theoretische Grundlagen der Informatik

Modul: 12060 Datenstrukturen und Algorithmen

2. Modulkürzel:	051510005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	6.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Erhard Plödereder
9. Dozenten:	Stefan Funke
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 2. Semester → Basismodule BA (Komb) Informatik, 2. Semester → Module im Nebenfach
11. Voraussetzungen:	• Modul 051520005 Programmierung und Software-Entwickung
12. Lernziele:	Die Studierenden kennen nach engagierter Mitarbeit in dieser Veranstaltung diverse zentrale Algorithmen auf geeigneten Datenstrukturen, die für eine effiziente Nutzung von Computern unverzichtbar sind. Sie können am Ende zu gängigen Problemen geeignete programmiersprachliche Lösungen angeben und diese in einer konkreten Programmiersprache formulieren. Konkret: • Kenntnis der Eigenschaften elementarer und häufig benötigter Algorithmen • Verständnis für die Auswirkungen theoretischer und tatsächlicher Komplexität • Erweiterung der Kompetenz im Entwurf und Verstehen von Algorithmen und der zugehörigen Datenstrukturen • Erste Begegnung mit nebenläufigen Algorithmen; sowohl „originär“ parallel, als auch parallelisierte Versionen bereits vorgestellter sequentieller Algorithmen
13. Inhalt:	• Vorgehensweise bei der Entwicklung und Implementierung von Algorithmen • Komplexität und Effizienz von Algorithmen, O-Notation • Wahl der Datenstrukturen; Listen, Bäume, Graphen; deren Definitionen, deren Datenstrukturen • diverse interne und externe Such- und Sortierverfahren (z.B. Linear-, Binär-, Interpolationssuche, AVL-, B-Bäume, internes und externes Hashing, mehrere langsame Sortierungen, Heap-, Quick-, Bucket-, Mergesort) • diverse Graphenalgorithmen (DFS, BFS, Besuchssequenzen, topol. Traversierung, Zusammenhangskomponenten, minimale Spannbäume, Dijkstra-, Floyd- kürzeste Wege) • Algorithmen auf Mengen und Relationen (transitive Hüllen, Warshall) • Korrektheitsbegriff und -formalismen; Spezifikation und Implementierung • Einige parallele und parallelisierte Algorithmen • einfache Elemente paralleler Programmierung, soweit für obiges notwendig
14. Literatur:	• Appelrath H.J., Ludewig. J., Skriptum Informatik, 1999 • Sedgewick, R., Algorithms in C, 1998
15. Lehrveranstaltungen und -formen:	• 120601 Vorlesung Datenstrukturen und Algorithmen

	• 120602 Übung Datenstrukturen und Algorithmen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 63 Stunden Nachbearbeitungszeit: 207 Stunden
17. Prüfungsnummer/n und -name:	12061 Datenstrukturen und Algorithmen (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein. Die genauen Details der Übungsleistungen und Ihrer Anrechnung werden zu Beginn der Lehrveranstaltung bekannt gegeben.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Mathematik, 2. Semester → Nebenfach → Nebenfach Informatik B.Sc. Technische Kybernetik, 2. Semester → Ergänzungsmodule → Höhere Informatik B.Sc. Softwaretechnik, 2. Semester → Basismodule B.Sc. Wirtschaftsinformatik, 2. Semester → Informatik (B 1) B.Sc. Mechatronik, 2. Semester → Kernmodule B.Sc. Maschinelle Sprachverarbeitung, 2. Semester → Basismodule B.Sc. Technikpädagogik, 2. Semester → Hauptfach Informatik → Basismodule Informatik B.Sc. Technikpädagogik, 2. Semester → Wahlpflichtfach → Informatik M.Sc. Technikpädagogik, 2. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Basismodule Allgemein Lehramt (GymPO I) ab PO 2010 → Pflichtmodule

Modul: 10190 Mathematik für Informatiker und Softwaretechniker

2. Modulkürzel:	080300100	5. Moduldauer:	2 Semester
3. Leistungspunkte:	18.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	12.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Wolfgang Rump
9. Dozenten:	Wolfgang Rump
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 1. Semester → Basismodule
11. Voraussetzungen:	Keine, die Teilnahme an einem Mathematik Vorkurs wird empfohlen.
12. Lernziele:	Die Studierenden haben die mathematischen Grundlagen für die Studiengänge Informatik bzw. Softwaretechnik erarbeitet und den selbständigen und kreativen Umgang mit den mathematischen Stoffgebieten gelernt.
13. Inhalt:	1. Semester: • Grundlagen (Aussagenlogik, Mengen, Relationen, Abbildungen, Zahlenmengen, Grundbegriffe der Algebra) • Lineare Algebra (Vektorräume, lineare Abbildungen, Matrizen, Determinanten, lineare Gleichungssysteme, Eigenwerte, Normalformen, Hauptachsentransformation, Skalarprodukte) • Analysis (Konvergenz, Zahlenfolgen und Zahlenreihen, stetige Abbildungen, Folgen und Reihen von Funktionen, spezielle Funktionen) 2. Semester: • Differential- und Integralrechung (Funktionen einer und mehrerer Variablen, Ableitungen, Taylorentwicklungen, Extremwerte, Integration, Anwendungen) • Gewöhnliche Differentialgleichungen (elementar lösbare Differentialgleichungen, Existenz und Eindeutigkeit von Lösungen)
14. Literatur:	• Anna Sändig, Mathematik, Vorlesungskripte , SS 2007 • D. Hachenberger, Mathematik für Informatiker, 2005 • M. Brill, Mathematik für Informatiker, 2001 • P.Hartmann, Mathematik für Informatiker, 2002
15. Lehrveranstaltungen und -formen:	• 101901 Vorlesung Mathematik 1 für Informatik und Softwaretechnik • 101902 Übung Mathematik 1 für Informatik und Softwaretechnik • 101903 Vorlesung Mathematik 2 für Informatik und Softwaretechnik • 101904 Übung Mathematik 2 für Informatik und Softwaretechnik
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 126 Stunden Nachbearbeitungszeit: 414 Stunden
17. Prüfungsnummer/n und -name:	10191 Mathematik für Informatiker und Softwaretechniker (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Ein Übungsschein aus den beiden Veranstaltungen, jeweils im 1. oder 2. Fachsemester zu erwerben
18. Grundlage für ... :	
19. Medienform:	

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 1. Semester → Basismodule B.Sc. Technikpädagogik, 1. Semester → Hauptfach Informatik → Basismodule Informatik Allgemein Lehramt (GymPO I) ab PO 2010 → Pflichtmodule

Modul: 10260 Programmierkurs

2. Modulkürzel:	051520010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	2.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Jochen Ludewig
9. Dozenten:	• Sascha Riexinger • Holger Röder
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 1. Semester → Basismodule BA (Komb) Informatik, 1. Semester → Module im Nebenfach
11. Voraussetzungen:	Keine
12. Lernziele:	Selbstständiges Erstellen von Programmen und Lösung von Programmieraufgaben in einer vorgegebenen Programmiersprache wie Java.
13. Inhalt:	Der Programmierkurs ergänzt die Vorlesung Programmierung und Software-Entwicklung (PSE). Die Teilnehmer erlernen eine weitere Programmiersprache (Java). Ihre Merkmale, Syntax und Semantik, werden denen der in PSE gelehrten Sprache gegenübergestellt. Praktische Übungen bereiten die Teilnehmer auf die Bearbeitung der Schein-Aufgabe vor. Die Lehrveranstaltung findet in zwei Varianten statt. Die Teilnahme richtet sich nach dem Studiengang: S. Riexinger: • BSc. Informatik • BA (Komb) Informatik • BSc. Maschinelle Sprachverarbeitung H. Röder: • BSc. Softwaretechnik • BSc. Wirtschaftsinformatik • BSc. Technikpädagogik • MSc. Technikpädagogik
14. Literatur:	
15. Lehrveranstaltungen und -formen:	102601 Übung Programmierkurs
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 21 Stunden Nachbearbeitungszeit: 69 Stunden
17. Prüfungsnummer/n und -name:	10261 Programmierkurs (USL), Sonstiges, Gewichtung: 1.0, Übungsschein - Scheinkriterien werden zu Beginn der Veranstaltung angekündigt.
18. Grundlage für ... :	
19. Medienform:	

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

B.Sc. Softwaretechnik, 1. Semester
→ Basismodule

B.Sc. Wirtschaftsinformatik, 1. Semester
→ Informatik (B 1)

B.Sc. Maschinelle Sprachverarbeitung, 1. Semester
→ Basismodule

B.Sc. Technikpädagogik, 1. Semester
→ Hauptfach Informatik
→ Basismodule Informatik

B.Sc. Technikpädagogik, 1. Semester
→ Wahlpflichtfach
→ Informatik

M.Sc. Technikpädagogik, 1. Semester
→ Wahlpflichtfach B
→ Affines Wahlpflichtfach Informatik
→ Affines Wahlpflichtfach Informatik Basismodule

Modul: 10280 Programmierung und Software-Entwicklung

2. Modulkürzel:	051520005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	6.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Bernhard Mitschang
9. Dozenten:	Bernhard Mitschang
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 1. Semester → Basismodule BA (Komb) Informatik, 1. Semester → Module im Nebenfach
11. Voraussetzungen:	Keine. Teilnahme an einem Mathematik Vorkurs wird empfohlen.
12. Lernziele:	Die Teilnehmer haben die wichtigsten Konzepte einer höheren Programmiersprache und ihrer Verwendung verstanden und sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen) zu analysieren und selbst zu konzipieren und zu implementieren. Sie kennen die Möglichkeiten, Daten- und Ablaufstrukturen zu entwerfen, zu beschreiben und zu codieren. Sie haben die Abstraktionskonzepte moderner Programmiersprachen verstanden. Sie kennen die Techniken und Notationen zur Definition kontextfreier Programmiersprachen und können damit arbeiten.
13. Inhalt:	• Algorithmen und funktionale Programmierung Kap. 1 verwendet nur die funktionale Teilmenge der Programmiersprache Ada, keine Variablen, keine Prozeduren. Grammatik, Formale Sprachen und BNF werden eingeführt. • Imperative Programmierung Kap. 2 erweitert die verwendete Sprache durch die prozeduralen Konzepte, also Variablen und Prozeduren. Zu den Sprachkonstrukten werden Vor- und Nachbedingungen, mit den Schleifen die Invarianten eingeführt. Datentypen werden schrittweise ausgebaut. In Zusammenhang mit den Zeigern werden die Konzepte für Keller und Halde vermittelt. Die Entwicklung einfacher Programme wird gezeigt und geübt. • Aufbau und Organisation komplexer Programme. Die Modularisierung, die bei größeren Programmen notwendig ist, führt zur Kapselung und zu den abstrakten Datentypen. Damit entsteht die Möglichkeit, neue Datenstrukturen und Datentypen sicher zu definieren. Die Konzepte der Kompilation und der Interpretation werden erläutert. Wichtige Beispiele komplexer Datentypen werden entwickelt. Die Konzepte der Generalisierung (generische Einheiten) werden vermittelt. • Ausnahmebehandlung Möglichkeiten und Probleme der Ausnahmebehandlung sind Gegenstände dieses kurzen Kapitels. • Objektorientierte Programmierung Am Ende des Semesters steht ein Ausblick in die objektorientierte Programmierung, d.h. die Umsetzung der bereits bekannten Konzepte (ADTs) in die objektorientierte Sichtweise und die Vererbung. Dieser Teil bereitet die Programmierung in einer objektorientierten Sprache (3. Semester) vor.
14. Literatur:	• Manuskripte: V.Claus (WS 08/09 bis SS 2009)

	• Appelrath, Hans-Jürgen und Ludewig, Jochen, "Skriptum Informatik - eine konventionelle Einführung", Verlag der Fachvereine Zürich und B.G. Teubner Stuttgart, 4. Auflage 1999 • Nagl., M., "Softwaretechnik mit Ada 95. Entwicklung großer Systeme.", Vieweg-Verlag, Wiesbaden 1999 • Barnes, J.G.P., "Programming in Ada 95", 2. Auflage, Addison-Wesley 1998
15. Lehrveranstaltungen und -formen:	• 102801 Vorlesung Programmierung und Softwareentwicklung • 102802 Übung Programmierung und Softwareentwicklung
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 63 Stunden Vor-/Nachbearbeitungszeit: 187 Stunden Prüfungsvorbereitung: 20 Stunden
17. Prüfungsnummer/n und -name:	10281 Programmierung und Software-Entwicklung (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Studienleistung: Übungsschein. Voraussetzungen werden zu Beginn vom Dozenten festgesetzt. Dazu gehören eine bestimmte Anzahl von Vorträgen in den Übungen und ein bestimmter Teil der Übungspunkte. Modulprüfung: schriftlich, 120 Minuten, keine Hilfsmittel
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 1. Semester → Basismodule B.Sc. Wirtschaftsinformatik, 1. Semester → Informatik (B 1) B.Sc. Mechatronik, 1. Semester → Kernmodule B.Sc. Maschinelle Sprachverarbeitung, 1. Semester → Basismodule B.Sc. Technikpädagogik, 1. Semester → Hauptfach Informatik → Basismodule Informatik B.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach → Informatik M.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Basismodule B.Sc. Simulation Technology, 1. Semester → Grundstudium Allgemein Lehramt (GymPO I) ab PO 2010, 1. Semester → Pflichtmodule

Modul: 10930 Technische Grundlagen der Informatik

2. Modulkürzel:	051711005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	6.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Martin Radetzki
9. Dozenten:	• Martin Radetzki • Sven Simon
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 1. Semester → Basismodule
11. Voraussetzungen:	Keine
12. Lernziele:	• Elektrotechnische Grundlagen: Der Studierende hat grundlegendes Verständnis elektrischer Schaltkreise, der Funktionsweise der Bauelemente und Komponenten von Computer-Systemen, wie Transistoren, Halbleiterschaltungen, RAM, ROM, Festplatte etc. erworben. • Digitaltechnische Komponenten: Der Studierende kann digitale Schaltungen von begrenzter Komplexität analysieren, konstruieren und optimieren.
13. Inhalt:	Elektrotechnische Grundlagen: • Elektrische Grundgrößen, Ohmsches Gesetz, Kirchhoffsche Gesetze. • Bauelemente: Widerstand, Kondensator, Spule, Bauelemente, Halbleiter-Leitungsmechanismen. • CMOS-Transistoren. • Integrationstechniken der Mikroelektronik. • Digitale Grundschaltungen, Logik- und Speicherschaltungen. • Technologie und Schaltungstechnik • Mikroprozessoren, Mikrocontroller, Signalprozessoren, FPGA. Digitaltechnische Komponenten: • Schaltalgebra, Schaltnetze / kombinatorische Netzwerke, • Modelle sequentiellen Verhaltens, • Schaltwerke / sequentielle Netzwerke, • Verzögerungsanalyse, • Taktschemata, • Binäre Codierung, • Datenpfadelemente, • Entwurfsmethodik und Entwurfsautomatisierung
14. Literatur:	-
15. Lehrveranstaltungen und -formen:	• 109301 Vorlesung Elektrotechnische Grundlagen • 109302 Übung Elektrotechnische Grundlagen • 109303 Vorlesung Digitaltechnische Komponenten • 109304 Übung Digitaltechnische Komponenten
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 63 Stunden Nachbearbeitungszeit: 117 Stunden

17. Prüfungsnummer/n und -name:	10931 Technische Grundlagen der Informatik (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Teilnahme an einer Mindestzahl der Übungen, die zu Beginn der Lehrveranstaltung festgelegt wird
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Technikpädagogik, 1. Semester → Hauptfach Informatik → Kernmodule Informatik M.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Kernmodule

Modul: 10940 Theoretische Grundlagen der Informatik

2. Modulkürzel:	050420005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	8.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Volker Diekert
9. Dozenten:	• Ulrich Hertrampf • Volker Diekert
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 1. Semester → Basismodule
11. Voraussetzungen:	Keine
12. Lernziele:	• Logik und Diskrete Strukturen: Die Studierenden haben die grundsätzlichen Kenntnisse in Logik und Diskreter Mathematik erworben, wie sie in den weiteren Grundvorlesungen der Informatik in verschiedenen Bereichen benötigt werden. • Automaten und Formale Sprachen: Die Studierenden beherrschen wichtige theoretische Grundlagen der Informatik, insbesondere die Theorie und Algorithmik endlicher Automaten. Hierzu gehört das Kennenlernen, Einordnung und Trennung der Chomskyschen Sprachklassen.
13. Inhalt:	• Logik und Diskrete Strukturen: Einführung in die Aussagenlogik; formale Sprache; Semantik (Wahrheitswerte); Syntax (Axiome und Schlussregeln); Normalformen; Hornformeln; aussagenlogische Resolution; Korrektheit und Vollständigkeit für die Aussagenlogik; Einführung in die Prädikatenlogik 1. Stufe; formale Sprache; Semantik und Syntax; Normalformen; Herbrand-Theorie; prädikatenlogische Resolution; Kombinatorik, Graphen, elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, RSA-Verfahren. • Automaten und Formale Sprachen: Deterministische- bzw. nichtdeterministische endliche Automaten, reguläre Ausdrücke, Minimierung endlicher Automaten, Iterationslemmata für reguläre und kontextfreie Sprachen, Normalformen, Kellerautomaten, Lösen des Wortproblems kontextfreier Sprachen mit dem CYK-Algorithmus, linear beschränkte Automaten, kontextsensitive Grammatiken, Typ 0-Grammatiken und Turingmaschinen.
14. Literatur:	• John Hopcroft, Jeffrey Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988 • Uwe Schöning, Theoretische Informatik - kurzgefasst, 1999
15. Lehrveranstaltungen und -formen:	• 109401 Vorlesung Logik und Diskrete Strukturen • 109402 Übung Logik und Diskrete Strukturen • 109403 Vorlesung Automaten und Formale Sprachen • 109404 Übung Automaten und Formale Sprachen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 84 Stunden

	Nachbearbeitungszeit: 276 Stunden
17. Prüfungsnummer/n und -name:	10941 Theoretische Grundlagen der Informatik (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 1. Semester → Basismodule B.Sc. Maschinelle Sprachverarbeitung, 1. Semester → Basismodule B.Sc. Technikpädagogik, 1. Semester → Hauptfach Informatik → Kernmodule Informatik B.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach → Informatik M.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Kernmodule Allgemein Lehramt (GymPO I) ab PO 2010, 1. Semester → Pflichtmodule

200 Kernmodule

Zugeordnete Module:

10020	Algorithmik
14910	Berechenbarkeit und Komplexität
25610	Grundlagen des Software Engineerings
10210	Mensch-Computer-Interaktion
10220	Modellierung
10240	Numerische und Stochastische Grundlagen
10270	Programmierparadigmen
10310	Rechnerorganisation
10330	Systemkonzepte und -programmierung

Modul: 10020 Algorithmik

2. Modulkürzel:	050420015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Volker Diekert
9. Dozenten:	• Ulrich Hertrampf • Volker Diekert
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 5. Semester → Kernmodule
11. Voraussetzungen:	Grundvorlesungen in theoretischer und praktischer Informatik.
12. Lernziele:	• Kennenlernen und beherrschen wichtiger Programmierparadigmen und Entwurfsstrategien; • Selbstständiges Erarbeiten von Laufzeitabschätzungen.
13. Inhalt:	• Entwurfsstrategien für Algorithmen (Teile und Beherrsche, Gierige Methode, Dynamische Programmierung, Backtracking, heuristische Algorithmen) • Analyse und Komplexität von Algorithmen • Mustererkennung • Sortierverfahren und ihre Komplexität • Verwaltung von Mengen • Union-Find-Algorithmen • Konvexe Hülle • optimale (Teil-) Bäume • Minimale Schnitte • Randomisierte Algorithmen und weitere Themen.
14. Literatur:	• Alfred V. Aho, John E. Hopcroft, Jeffrey of Computer Algorithms, 1974 • Alfred V. Aho, John E. Hopcroft, Jeffrey Algorithms, 1987 • T. Ottmann und P. Widmayer, Algorithmen 2004 • Thomas H. Cormen, Charles E. Leiserson, Introduction to Algorithms (Second Edition), • Volker Diekert, Entwurf und Analyse effizienter (Vorlesungsskript), 2006
15. Lehrveranstaltungen und -formen:	• 100201 Vorlesung Algorithmik • 100202 Übung Algorithmik
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10021 Algorithmik (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvoraussetzung: Übungsschein
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

21. Zuordnung zu weiteren Curricula:	B.Sc. Wirtschaftsinformatik, 5. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich II: Informatik Basis

Modul: 14910 Berechenbarkeit und Komplexität

2. Modulkürzel:	050420010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Volker Diekert
9. Dozenten:	• Stefan Funke • Volker Diekert • Ulrich Hertrampf
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Kernmodule
11. Voraussetzungen:	Inhaltliche Voraussetzungen: Theoretische Grundlagen der Informatik, Mathematik für Informatiker 1 und 2 (abgedeckt durch Pflichtmodule im Grundstudium).
12. Lernziele:	Die Teilnehmer beherrschen wichtige theoretische Grundlagen der Informatik, können Probleme in Kategorien einordnen wie entscheidbar/ unentscheidbar, effizient lösbar, deterministische/nichtdeterministische Berechnungen.
13. Inhalt:	Gleichwertigkeit der verschiedenden Konkretisierungen des Algorithmenbegriffs, Churchsche These, Grenzen zwischen Entscheidbarbkeit und Unentscheidbarkeit. Turing-Berechenbarkeit, primitiv-rekursive Funktionen, mu-rekursive Funktionen, Halteproblem, Satz von Rice, Gödelscher Satz. Wichtige Komplexitätsklassen, P-NP-Problem, NP-Vollständigkeit, Satz von Cook.
14. Literatur:	• Christos H. Papadimitriou, Computational Complexity , 1994 • John E. Hopcroft, Jeffrey D. Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988 • Volker Diekert, Komplexitätstheorie (Vorlesungsskript), 2007
15. Lehrveranstaltungen und -formen:	• 149101 Vorlesung Berechenbarkeit und Komplexität • 149102 Übung Berechenbarkeit und Komplexität
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42h Nachbearbeitungszeit: 118h Prüfungsvorbereitung: 20h **Gesamt: 180h**
17. Prüfungsnummer/n und -name:	14911 Berechenbarkeit und Komplexität (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :	10020 Algorithmik
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Mathematik, 5. Semester → Ergänzungsmodule

B.Sc. Simulation Technology, 3. Semester
→ Fachstudium
→ Vertiefungsrichtung CS

Allgemein Lehramt (GymPO I) ab PO 2010
→ Wahlmodule

Modul: 25610 Grundlagen des Software Engineerings

2. Modulkürzel:	51520170	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Stefan Wagner (ISTE)
9. Dozenten:	Stefan Wagner
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Kernmodule
11. Voraussetzungen:	051520005 Programmierung und Software-Entwicklung 051510005 Datenstrukturen und Algorithmen sowie entsprechende Programmiererfahrung
12. Lernziele:	Die Teilnehmer kennen die Grundbegriffe der Softwaretechnik und haben einen Überblick über die Methoden und Techniken, die im Software Engineering angewandt werden.
13. Inhalt:	GSE vermittelt Grundlagen der Softwaretechnik. Dieses Modul kommt auch, wenn die Voraussetzungen erfüllt sind, für andere Fachrichtungen in Frage. Die Vorlesung behandelt technische und andere Aspekte der Softwarebearbeitung. Die einzelnen Themen sind: • Geschichte und Konzepte des Software Engineerings • Software-Management; Software-Prüfung und Qualitätssicherung • Methoden, Sprachen und Werkzeuge für die einzelnen Phasen: Spezifikation, Grobentwurf, Feinentwurf, Codierung, Test
14. Literatur:	• Ludewig, Lichter: Software Engineering. dpunkt-Verlag, Heidelberg. 2. Aufl. 2010, • Pfleeger, Atlee: Software Engineering, Pearson. 2010
15. Lehrveranstaltungen und -formen:	• 256101 Vorlesung Grundlagen des Software Engineerings • 256102 Übung Grundlagen des Software Engineerings
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138
17. Prüfungsnummer/n und -name:	25611 Grundlagen des Software Engineerings (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Technische Kybernetik → Ergänzungsmodule → Höhere Informatik

Modul: 10210 Mensch-Computer-Interaktion

2. Modulkürzel:	051900001	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Albrecht Schmidt
9. Dozenten:	• Albrecht Schmidt • Thomas Ertl • Daniel Weiskopf
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Kernmodule
11. Voraussetzungen:	• 051520005 Programmierung und Software-Entwicklung • 051200005 Systemkonzepte und -programmierung
12. Lernziele:	Studierende entwickeln ein Verständnis für Modelle, Methoden und Konzepte der Mensch-Computer-Interaktion. Sie lernen verschiedene Ansätze für den Entwurf, die Entwicklung und Bewertung von Benutzungsschnittstellen kennen und verstehen deren Vor- und Nachteile.
13. Inhalt:	Die Vorlesung vermittelt Konzepte, Prinzipien, Modelle, Methoden und Techniken für die effektive Entwicklung von benutzerfreundlichen Mensch-Computer-Schnittstellen. Das Thema moderner Benutzungsschnittstellen wird dabei für klassische Computer aber auch für mobile Geräte, eingebettete Systeme, Automobile und intelligente Umgebungen betrachtet. Die folgenden Themen werden in der Vorlesung behandelt: • Einführung in die Grundlagen der Mensch-Computer Interaktion, historische Entwicklung • Entwurfsprinzipien und Modelle für moderne Benutzungsschnittstellen und interaktive Systeme • Informationsverarbeitung des Menschen, Wahrnehmung, Motorik, Eigenschaften und Fähigkeiten des Benutzers • Interaktionskonzepte und -stile, Metaphern, Normen, Regeln und Style Guides • Ein- und Ausgabegeräte, Entwurfsraum für interaktive Systeme • Analyse-, Entwurfs- und Entwicklungsmethoden und -werkzeuge für Benutzungsschnittstellen • Prototypische Realisierung und Implementierung von interaktiven Systemen, Werkzeuge • Architekturen für interaktive Systeme, User Interface Toolkits und Komponenten • Akzeptanz, Evaluationsmethoden und Qualitätssicherung
14. Literatur:	• Bernhard Preim, Raimund Dachselt. Interaktive Systeme 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung. Springer, Berlin; 2. Auflage. 2010 • Alan Dix, Janet Finley, Gregory Abowd, Russell Beale, Human-Computer Interaction, 2004 • Ben Shneiderman, Catherine Plaisant, Designing the User Interfaces, 2005

15. Lehrveranstaltungen und -formen:	• 102101 Vorlesung Mensch-Computer-Interaktion • 102102 Übung Mensch-Computer-Interaktion
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10211 Mensch-Computer-Interaktion (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog SWT B.Sc. Wirtschaftsinformatik, 6. Semester → Informatik (B 1) B.Sc. Wirtschaftsinformatik, 6. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich II: Informatik Basis B.Sc. Wirtschaftsinformatik, 6. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich III: Informatik Simulation B.Sc. Maschinelle Sprachverarbeitung, 5. Semester → Wahlbereich E/I

Modul: 10220 Modellierung

2. Modulkürzel:	052010001	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Frank Leymann
9. Dozenten:	• Bernhard Mitschang • Frank Leymann
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Kernmodule
11. Voraussetzungen:	• 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen • 051200005 Systemkonzepte und -programmierung
12. Lernziele:	Am Ende des Moduls sind die Studierenden in der Lage, wesentliche Artefakte eines IT Systems zu modellieren. Der Zusammenhang und das Zusammenspiel solcher Artefakte ist verstanden. Die Rolle von Metamodellen und deren Erstellung ist klar.
13. Inhalt:	• Entity-Relationship Modell & komplexe Objekte • Relationenmodell & Relationenalgebra , Überblick SQL • Transformationen von ER nach Relationen, Normalisierung • XML, DTD, XML-Schema, Info-Set, Namensräume • Metamodelle & Repository • RDF, RDF-S & Ontologien • UML • Petri Netze, Workflownetze • BPMN
14. Literatur:	• A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002 • R. Eckstein, S. Eckstein, "XML und Datenmodellierung", dpunkt.verlag 2004 • M. Hitz, G. Kappel, E. Kapsammer, W. Retschitzegger, UML @ Work - Objektorientierte Modellierung mit UML2, 2005 • P. Hitzler, M. Krötzsch, S. Rudolph, Y. Sure, Semantic Web, 2008 • T.J. Teorey, Database Modeling & Design, 2nd Edition, 1994 • H.J. Habermann, F. Leymann, "Repository", Oldenbourg 1993 • W. Reisig, "Petri-Netze", Vieweg & Teubner 2010 • B. Silver,"BPMN Method & Style",Cody-Cassidy Press 2009
15. Lehrveranstaltungen und -formen:	• 102201 Vorlesung Modellierung • 102202 Übung Modellierung
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10221 Modellierung (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :	• 10030 Architektur von Anwendungssystemen • 10080 Datenbanken und Informationssysteme
19. Medienform:	

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

B.Sc. Softwaretechnik, 4. Semester
- → Ergänzungsmodule
- → Katalog ISG

B.Sc. Wirtschaftsinformatik, 4. Semester
- → Informatik (B 1)

B.Sc. Technikpädagogik, 4. Semester
- → Hauptfach Informatik
- → Pflichtmodule Informatik

M.Sc. Technikpädagogik, 4. Semester
- → Wahlpflichtfach B
- → Affines Wahlpflichtfach Informatik
- → Affines Wahlpflichtfach Informatik Pflichtmodule

Modul: 10240 Numerische und Stochastische Grundlagen

2. Modulkürzel:	051240005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	6.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Marc Alexander Schweitzer
9. Dozenten:	• Stefan Zimmer • Michael Bader • Marc Alexander Schweitzer
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Kernmodule
11. Voraussetzungen:	• Modul 080300100 Mathematik für Informatiker und Softwaretechniker
12. Lernziele:	Beherrschung grundlegender Begriffe und Methoden der Numerik und Stochastik, Kenntnis der Anwendungsbereiche und Gültigkeitsgrenzen der erlernten Methoden, insbesondere Kenntnis der Auswirkungen von Näherungen, Beherrschung der Modellierung einfacher Probleme mit stochstischen Methoden.
13. Inhalt:	Methoden der angewandten Mathematik, insbesondere der Numerik, Stochastik und Statistik, sind für viele Bereiche der Informatik wie Simulation, Grafik oder Bildverarbeitung von zentraler Bedeutung. In Ergänzung der Mathematik-Grundausbildung vermittelt diese Vorlesung folgende Grundkenntnisse: • numerische Algorithmik • Gleitpunktzahlen und Gleitpunkarithmetik • Interpolation & Approximation • Integration • lineare Gleichungssysteme • Iterative Lösung linearer und nichtlinearer Gleichungen • gewöhnliche Differentialgleichungen • Stochastik • Zufall und Unsicherheit • diskrete und kontinuierliche Wahrscheinlichkeitsräume • Asymptotik • Elementare induktive Statistik Dabei wird ein konstruktiv-algorithmischer Zugang gewählt, der sich an konkreten Aufgabenstellungen aus der Informatik orientiert.
14. Literatur:	• Peter Bastian; Numerische und Stochastische Grundlagen der Informatik, 2008 • Huckle, Schneider; Numerik für Informatiker • Schickinger T., Steger A.; Diskrete Strukturen, Band 2, 2002 • Dahmen, Reusken; Numerik für Ingenieure
15. Lehrveranstaltungen und -formen:	• 102401 Vorlesung Numerische und Stochastische Grundlagen der Informatik • 102402 Übung Numerische und Stochastische Grundlagen der Informatik
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 63 Stunden Nachbearbeitungszeit: 207 Stunden

17. Prüfungsnummer/n und -name:	10241	Numerische und Stochastische Grundlagen (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:		
21. Zuordnung zu weiteren Curricula:		

Modul: 10270 Programmierparadigmen

2. Modulkürzel:	051510010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	2.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Erhard Plödereder
9. Dozenten:	• Viktor Avrutin • Erhard Plödereder
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Kernmodule
11. Voraussetzungen:	• Programmiererfahrung in einer ersten Programmiersprache • Modul 051520005 Programmierung und Softwareentwicklung • Modul 051520010 Programmierkurs • Modul 051510005 Datenstrukturen und Algorithmen
12. Lernziele:	Die Studierenden haben die Konzepte einer zweiten Programmiersprache ihrer Ausbildung verstanden und deren Grundbegriffe erlernt. Sie können ihre Kenntnisse in einfachen Programmen anwenden.
13. Inhalt:	Bezogen auf eine zweite Programmiersprache der Informatikausbildung behandelt die Lehrveranstaltung dieses Moduls die folgenden Inhalte: • Grundbegriffe und Techniken der zweiten Programmiersprache, insbesondere Aufbau und Struktur der Programme sowie Abstraktions- und Ausführungskonzepte der Sprache • Umsetzung der objekt-orientierten und der generischen Programmierparadigmen • Umgang mit Standardsituationen der Programmierung • Auswirkungen und Anwendungsbereiche der behandelten Programmierparadigmen Die Wahl der Programmiersprache wird vom Dozenten bekanntgegeben.
14. Literatur:	Java ist auch eine Insel, Christian Ullenboom, 1475 S., 8. Auflage, Galileo Computing 2009, ISBN 978-3-8362-1371-4, online-Version: http://openbook.galileocomputing.de/javainsel8/ Ken Arnold, James Gosling, David Holmes, The Java Programming Language, Fourth Edition, Addison-Wesley Professional, 2005, ISBN 0-321-34980-6
15. Lehrveranstaltungen und -formen:	102701 Übung Programmierparadigmen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 21 Stunden Nachbearbeitungszeit: 69 Stunden
17. Prüfungsnummer/n und -name:	10271 Programmierparadigmen (PL), Studienbegleitend, Gewichtung: 1.0, Studienbegleitende Abgabe von Programmierlösungen
18. Grundlage für ... :	
19. Medienform:	

20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Simulation Technology, 3. Semester → Fachstudium → Vertiefungsrichtung CS

Modul: 10310 Rechnerorganisation

2. Modulkürzel:	051700005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	9.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Hans-Joachim Wunderlich
9. Dozenten:	• Hans-Joachim Wunderlich • Martin Radetzki • Sven Simon
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Kernmodule
11. Voraussetzungen:	• Für Rechnerorganisation 1: Einführung in die Technische Informatik (14360) • Für Rechnerorganisation 2: Erfolgreiche Teilnahme an den Übungen von Rechnerorganisation 1 oder eine bestandene Eingangsklausur
12. Lernziele:	Rechnerorganisation 1: • Kenntnisse über die Grundlagen der Organisation von Rechnern und deren maschinennahe Programmierung • Gründzüge über die Beschreibung und den Entwurf von Hardwaresystemen Rechnerorganisation 2: • Kenntnisse in den Grundlagen des Entwurfs digitaler Schaltungen und Systeme, • Fertigkeiten in der Verwendung von Hardware-Beschreibungssprachen • Fertigkeiten im Umgang mit programmierbarer Logik (FPGA) und • Prototypenboards, • Fertigkeiten im Umgang mit Werkzeugen zur Entwurfsautomatisierung, • Kentnisse des Zusammenhangs von Hard- und Software • Erfahrung in Projektarbeit im Team
13. Inhalt:	Rechnerorganisation 1: • Aufbau, Entwurf und maschinennahe Programmierung von Mikroprozessoren werden erläutert. In den Übungen wird das Wissen durch schriftliche Aufgaben sowie Experimente mit Prozessorsimulatoren vertieft. Im einzelnen werden behandelt: • Grundstrukturen: Stack-, Akkumulator- und Register-basierende Rechner • Informationsdarstellung in Hardware: Daten- und Befehlsformate, Fehlererkennung und -korrektur, Befehlssätze und Unterstützung von Hochsprachen. • MIPS als RISC-Bespiel und seine Assemblerprogrammierung • Grundelemente und Entwurf auf Register-Transfer-Ebene • Grundzüge einer Hardware-Beschreibungssprache (VHDL) • Operationswerke: Multiplikation, Division, Gleitkommaeinheiten • Steuerwerksentwurf und Mikroprogrammierung • Befehlszyklus und Unterbrechungen • Pipelining und statisches Scheduling

	• Speicherorganisation: Cachestrukturen und virtueller Speicher, Seitenverwaltung, Segmentierung, TLB, MMU und DMA • Leistungsbewertung: Maßzahlen und CPI, Benchmarking und einfache Anwendung von Warteschlangen Rechnerorganisation 2: • Elementare Messtechnik • Aufbau wesentlicher Bauelemente und Grundschaltungen der Elektronik. • Entwurf eines einfachen RISC-Prozessors mit kommerziellen Entwurfssystemen. • Umsetzung in Hardware mit einem Field Programmable Gate Array (FPGA) und einem Prototypenboard. • Entwurf kombinatorischer und sequentieller Schaltungen. • Arbeitstechniken zur Komplexitätsbewältigung und Konzepte zur Schaltungsvalidierung. • Programmierung des selbst entworfenen Prozessors in Maschinensprache.
14. Literatur:	Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	• 103101 Vorlesung Rechnerorganisation 1 • 103102 Übung Rechnerorganisation 1 • 103103 Vorlesung Rechnerorganisation 2 • 103104 Hardwarepraktikum Rechnerorganisation 2
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 95 Stunden Nachbearbeitungszeit: 265 Stunden
17. Prüfungsnummer/n und -name:	• 10311 Rechnerorganisation (PL), schriftliche Prüfung, 90 Min., Gewichtung: 7.0 • 10312 Rechnerorganisation - Praktikum und Übungsaufgaben (LBP), schriftlich, eventuell mündlich, Gewichtung: 3.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	

Modul: 10330 Systemkonzepte und -programmierung

2. Modulkürzel:	051200005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Kurt Rothermel
9. Dozenten:	• Kurt Rothermel • Frank Leymann
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Kernmodule
11. Voraussetzungen:	• Modul 051520005 Programmierung und Software-Entwicklung • Modul 051510005 Datenstrukturen und Algorithmen
12. Lernziele:	• Verstehen grundlegender Architekturen und Organisationsformen von Software-Systemen • Verstehen systemnaher Konzepte und Mechanismen • Kann existierende Systemplattformen und Betriebssysteme hinsichtlich ihrer Eigenschaften analysieren und anwenden. • Kann systemnahe Software entwerfen und implementieren. • Kann nebenläufige Programme entwickeln • Kann mit Experten anderer Fachgebiete die Anwendung von Systemfunktionen abstimmen.
13. Inhalt:	Grundlegende Systemstrukturen - und organisationen • Multitaskingsystem • Multiprozessorsystem • Verteiltes System Modellierung und Analyse nebenläufiger Programme • Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm • Korrektheit- und Leitungskriterien Betriebssystemkonzepte • Organisation von Betriebssystemen • Prozesse und Threads • Eingabe/Ausgabe • Scheduling Konzepte zur Synchronisation über gemeinsamen Speicher • Synchronisationsprobleme und -lösungen • Synchronisationswerkzeuge: Semaphor, Monitor Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer • Taxonomie: Kommunikation und Synchronisation • Nachrichten als Kommunikationskonzept • Höhere Kommunikationskonzepte Basisalgorithmen für Verteilte Systeme • Erkennung globaler Eigenschaften • Schnappschussproblem • Konsistenter globaler Zustand • Verteilte Terminierung Praktische nebenläufige Programmierung in Java • Threads und Synchronisation • Socketschnittstelle • RMI Programmierung
14. Literatur:	• Literatur, siehe Webseite zur Veranstaltung

15. Lehrveranstaltungen und -formen:	• 103301 Vorlesung Systemkonzepte und -programmierung • 103302 Übung Systemkonzepte und -programmierung
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	• 10331 Systemkonzepte und -programmierung (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 7.0 • 10332 Systemkonzepte und -programmierung - Übungsschein (LBP), schriftlich, eventuell mündlich, Gewichtung: 3.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog SWT B.Sc. Technikpädagogik, 3. Semester → Hauptfach Informatik → Pflichtmodule Informatik M.Sc. Technikpädagogik, 3. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Pflichtmodule

300 Ergänzungsmodule

Zugeordnete Module:

320	Katalog ISG 1-3
330	Katalog ISW 1-3
340	Katalog ISW 4-7
350	Wahlmodule aus Master Informatik

320 Katalog ISG 1-3

Zugeordnete Module:

10140 Advanced Processor Architecture
10030 Architektur von Anwendungssystemen
10080 Datenbanken und Informationssysteme
10090 Grundlagen der Eingebetteten Systeme
10100 Grundlagen der Graphischen Ingenieursysteme
10110 Grundlagen der Künstlichen Intelligenz
10120 Grundlagen der Modellbildung und Simulation
10150 Grundlagen des Compilerbaus und der Programmiersprachen
10170 Imaging Science
10250 Parallele Systeme
11110 Verteilte Systeme

Modul: 10140 Advanced Processor Architecture

2. Modulkürzel:	051700010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch

8. Modulverantwortlicher:	Hans-Joachim Wunderlich
9. Dozenten:	• Hans-Joachim Wunderlich • Stefan Holst
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	• 051700005 Rechnerorganisation
12. Lernziele:	Good understanding of the basic concepts used in modern CPUs and computing systems. Awareness of the challenges in modern processor design and the reasoning behind current and future design trends.
13. Inhalt:	Classic topics in computer architecture as hardware/software interface are discussed as well as more advanced topics which include: • Technology basics: Design patterns, fabrication, yield, test and reliability, cost and quality, scaling. • Performance: Frequency and instructions per clock cycle, performance analysis and optimization. • Power dissipation: Analysis and optimization of power and performance, power and scaling. • Computer arithmetic: Efficient hardware for basic arithmetic, implementation of exponential, logarithm and trigonometric functions, floating point arithmetic and standards, arithmetic pipelines and filter, real-world floating point implementations like the Cell SPE or SPARC. • Instruction parallelism: Super scalar computing, static and dynamic scheduling, out-of-order execution, VLIW-processors, multithreading • Parallel architectures: Shared memory and message passing, multi-core processors, multi-core systems on a chip and emerging many-core technologies found in current graphic accelerators • Memory hierarchy: Memory technology and cache design. • Fault tolerance for single processors and multi processor systems
14. Literatur:	• I. Koren, Computer Arithmetic Algorithms, 2001 • J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 2006 • S. Iman, M. Pedram, Logic Synthesis for Low Power VLSI Designs, 1997
15. Lehrveranstaltungen und -formen:	• 101401 Vorlesung Grundlagen der Rechnerarchitektur

	• 101402 Übung Grundlagen der Rechnerarchitektur
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10141 Advanced Processor Architecture (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester → Ergänzendes Modul Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester → Wahlmodule

Modul: 10030 Architektur von Anwendungssystemen

2. Modulkürzel:	052010002	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Frank Leymann
9. Dozenten:	Frank Leymann
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	Vorlesungen des Grundstudiums.
12. Lernziele:	Am Ende des Moduls ist das Konzept einer Anwendungsarchitektur verstanden. Die Rolle von Middleware im Rahmen von Anwendungsarchitekturen ist klar. Grundsätzliche Strukturen und Muster in Anwendungsarchitekturen sind bekannt. Nicht-funktionale Eigenschaften von Anwendungssystemen und deren Bedeutung sind verstanden.
13. Inhalt:	• "Anwendungsarchitektur": Terminologie, Rollen & Artefakte • Datenbanksysteme • Schichtenarchitekturen (Client-Server Cut, N-Tier) • API, RPC & Middleware • Transaktionen • Nachrichtenorientierung • Message-Oriented Middleware • TP Monitore (direct TP, queued TP) • QoS (high availability, scalability, security,...) • Komponenten & ihre Beschreibung (EJB & WSDL) • Application Server (JEE) • Architekturstile (POSA,...) • Lose Kopplung & SOA
14. Literatur:	• A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002 • B. Neubauer, T. Ritter, F. Stoinnski, CORBA Komponenten, 2004 • F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, Pattern-orientierte Software Architektur - Ein Patternsystem, 1998 • F. Leymann, D. Roller, Production Workflow, 2000 • L. Hohmann, Beyond Software Architecture, 2003 • M. Fowler, Patters of Enterprise Application Architecture, 2003 • P. Bernstein, E. Newcomer, Principles of Transaction Processing, 1997 • S. Conrad, W. Hasselbring, A. Koschel, R. Tritsch, Enterprise Application Integration, 2006 • S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, Web Services Platform Architecture, 2005 • W. Emmerich, Konstruktion von verteilten Objekten, 2003

15. Lehrveranstaltungen und -formen:	• 100301 Vorlesung Grundlagen der Architektur von Anwendungssystemen • 100302 Übung Grundlagen der Architektur von Anwendungssystemen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10031 Architektur von Anwendungssystemen (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISG

Modul: 10080 Datenbanken und Informationssysteme

2. Modulkürzel:	051200025	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Bernhard Mitschang
9. Dozenten:	• Bernhard Mitschang • Holger Schwarz
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 5. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	• Modul 052010001 Modellierung
12. Lernziele:	Die Studierenden haben das erforderliche Verständnis und die Kenntnisse für die Implementierung von Datenbanksystemen erworben.
13. Inhalt:	Diese Vorlesung ist als Einstiegsvorlesung für das Vertiefungsgebiet Datenbanksysteme konzipiert. Es wird dabei versucht, aufbauend auf dem Inhalt der Vorlesung Modellierung, die wesentlichen Themen aus dem Bereich Datenbanksysteme zu vertiefen, insbesondere unter Berücksichtigung von Implementierungskonzepten und -techniken. Übersicht zur Stoffauswahl: • Datenbanksystemarchitektur und Implementierungskonzepten • Implementierung relationaler Anfragesprachen (Anfrageverarbeitung und Anfrageoptimierung) • Transaktionsverarbeitung Stoffauswahl, -umfang und Detaillierungsgrad werden aus der Sicht der Datenbanksystemimplementierung getroffen.
14. Literatur:	• A. Kemper, A. Eickler, Datenbanksysteme - Eine Einführung, 2004 • T. Härder, E. Rahm, Datenbanksysteme - Konzepte und Techniken der Implementierung, 2001 • H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems. The Complete Book, 2003 • R. Elmasri, S. Navathe, Fundamentals of Database Systems, 2003
15. Lehrveranstaltungen und -formen:	• 100801 Vorlesung Datenbanken und Informationssysteme • 100802 Übung Datenbanken und Informationssysteme
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10081 Datenbanken und Informationssysteme (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Modalitäten werden in der ersten Vorlesung angegeben
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Maschinelle Sprachverarbeitung, 5. Semester → Wahlbereich E/I Allgemein Lehramt (GymPO I) ab PO 2010, 5. Semester → Ergänzendes Modul Allgemein Lehramt (GymPO I) ab PO 2010, 5. Semester → Wahlmodule

Modul: 10090 Grundlagen der Eingebetteten Systeme

2. Modulkürzel:	051711010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Martin Radetzki
9. Dozenten:	Martin Radetzki
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	• Modul 051700005 Rechnerorganisation
12. Lernziele:	Methodische und werkzeugunterstützte Konstruktion von Hardware und Software für auf spezifische Anwendungen hin optimierte eingebettete Systeme.
13. Inhalt:	Modelle zur Systemspezifikation, Modellierungssprachen und ihre Simulation, System- und Architektursynthese, Allokation von Ressourcen und Bindung von Aufgaben/Operationen, Hardware-Software-Partitionierung, Verfahren zur Ablaufplanung für parallele Architekturen, Optimierungsverfahren, Anwendungsspezifische Prozessoren, On-Chip/ Board-Verbindungsnetzwerke.
14. Literatur:	• J. Teich, Digitale Hardware/Software-Systeme, 2. Auflage, 2007 • P. Marwedel, Embedded System Design, 2006
15. Lehrveranstaltungen und -formen:	• 100901 Vorlesung Grundlagen der Eingebetteten Systeme • 100902 Übung Grundlagen der Eingebetteten Systeme
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	• 10091 Grundlagen der Eingebetteten Systeme (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 75.0 • 10092 Grundlagen der Eingebetteten Systeme - Rechnerübungen (LBP), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 25.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW

Modul: 10100 Grundlagen der Graphischen Ingenieursysteme

2. Modulkürzel:	051400005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Dieter Roller
9. Dozenten:	Dieter Roller
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	Grundstudium
12. Lernziele:	• Kenntnis und Verständnis von Modellen bei der Produktentwicklung • Grundkenntnisse über die wichtigsten Modellarten, Algorithmen und Datenstrukturen und Techniken für den Datenaustausch
13. Inhalt:	Inhalte: • Anforderungen an CAD-Systeme • zweidimensionale Modelle • dreidimensionale Modelle • interaktive Modellerstellung • Einführung in die Modifikationstechnik u. parametrische Modellierung • Methoden zur Modellmodifikation • Grundlagen der parametrischen Modellierung • Ansätze und Verfahren zur parametrischen Variantenerzeugung • Ausgewählte Anwendungsbeispiele • Überblick über weitergehende Modellieransätze • Datenverwaltung in CAD
14. Literatur:	• D. Roller, CAD - Effiziente Anpassungs- und Variantenkonstruktion, Springer-Verlag • Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	• 101001 Vorlesung Grundlagen der Graphischen Ingenieursysteme • 101002 Übung Grundlagen der Graphischen Ingenieursysteme
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10101 Grundlagen der Graphischen Ingenieursysteme (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule

→ Katalog ISW

Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
→ Ergänzendes Modul

Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
→ Wahlmodule

Modul: 10110 Grundlagen der Künstlichen Intelligenz

2. Modulkürzel:	051900205	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Gunther Heidemann
9. Dozenten:	Gunther Heidemann
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	• Modul 080300100 Mathematik für Informatiker und Softwaretechniker
12. Lernziele:	Der Student / die Studentin beherrscht die Grundlagen der Künstlichen Intelligenz, kann Probleme der KI selbständig einordnen und mit den erlernten Methoden und Algorithmen bearbeiten.
13. Inhalt:	• Intelligenz • Agentenbegriff • Problemlösen durch Suchen, Suchverfahren • Probleme mit Rand- und Nebenbedingungen • Spiele • Aussagen- und Prädikatenlogik • Logikbasierte Agenten, Wissensrepräsentation • Inferenz • Planen • Unsicherheit, probabilistisches Schließen • Probabilistisches Schließen über die Zeit • Sprachverarbeitung • Entscheidungstheorie • Lernen
14. Literatur:	• G. F. Luger, Künstliche Intelligenz, 2001 • S. Russell, P. Norvig, Künstliche Intelligenz, 2004
15. Lehrveranstaltungen und -formen:	• 101101 Vorlesung Grundlagen der Künstlichen Intelligenz • 101102 Übung Grundlagen der Künstlichen Intelligenz
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10111 Grundlagen der Künstlichen Intelligenz (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG

B.Sc. Softwaretechnik, 5. Semester
→ Ergänzungsmodule
→ Katalog ISW

B.Sc. Wirtschaftsinformatik, 5. Semester
→ Informatik (B 1)

B.Sc. Wirtschaftsinformatik, 5. Semester
→ Wahlpflichtbereich (Bereich C)
→ Wahlbereich II: Informatik Basis

B.Sc. Wirtschaftsinformatik, 5. Semester
→ Wahlpflichtbereich (Bereich C)
→ Wahlbereich III: Informatik Simulation

B.Sc. Maschinelle Sprachverarbeitung, 5. Semester
→ Wahlbereich E/I

Modul: 10120 Grundlagen der Modellbildung und Simulation

2. Modulkürzel:	051240010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Marc Alexander Schweitzer
9. Dozenten:	• Stefan Zimmer • Michael Bader • Marc Alexander Schweitzer
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	• 080300100 Mathematik für Informatiker und Softwaretechniker • 051240005 Numerische und Stochastische Grundlagen der Informatik
12. Lernziele:	Beherrschung des grundsätzlichen Vorgehens in der Modellbildung. Kenntnis einer Auswahl diskreter und kontinuierlicher Modelle und entsprechender Simulationsmethoden. Fähigkeit, mit den erlernten Kenntnissen selbständig numerische Methoden problemorientiert um- und einzusetzen.
13. Inhalt:	Diese Vorlesung bietet eine Einführung in die Grundlagen der Modellbildung und Simulation mit dem Ziel der Vorbereitung auf weiterführende Vorlesungen in diesem Bereich. Da Simulationsmethoden oft für viele verschiedene Problemklassen einsetzbar sind, ist die Vorlesung methodisch strukturiert. Den Hauptteil der Vorlesung bilden hierbei kontinuierliche Modelle sowie deren numerische Behandlung. Vorgestellt werden insbesondere die Diskretisierung gewöhnlicher und partieller Differentialgleichungen mit Finiten Differenzen sowie die Lösung großer dünnbesetzter Gleichungssysteme, Eigenwert- und Minimierungsprobleme. Diese Verfahren werden auf Problemstellungen der Natur- und Ingenieurwissenschaften angewendet, z.B. Populationwachstum, Wärmetransport und Verkehrssimulation.
14. Literatur:	• Modellbildung und Simulation - Eine anwendungsorientierte Einführung; Bungartz, H.-J., Zimmer, S., Buchholz, M., Pflüger, D. , Springer Verlag, eXamen.press, 2009, ISBN 978-3-540-79809-5
15. Lehrveranstaltungen und -formen:	• 101201 Vorlesung Grundlagen der Modellbildung und Simulation • 101202 Übung Grundlagen der Modellbildung und Simulation
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10121 Grundlagen der Modellbildung und Simulation (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG
	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW
	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog SWT
	Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester → Ergänzendes Modul
	Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester → Wahlmodule

Modul: 10150 Grundlagen des Compilerbaus und der Programmiersprachen

2. Modulkürzel:	051510015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Erhard Plödereder
9. Dozenten:	Erhard Plödereder
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	Grundkenntnisse aus den Einführungsvorlesungen des Informatikgrundstudiums, sowie einige Erfahrungen mit Programmierung. Vorkenntnisse über formale Sprachen sind vorteilhaft, aber nicht zwingend.
12. Lernziele:	Die Studierenden haben die Grundkenntnisse erlangt, die zur effizienten Verwendung von Lexer- und Parser-Generatoren zur Analyse von Eingabetexten nötig sind. Sie haben gelernt, die Fehlermeldungen aus diesen Generatoren und den Compilern oder Interpretern richtig einzuordnen. Ferner haben sie durch Betrachtung der Implementierungsmodelle typischer Programmiersprachenkonstrukte Verständnis für das Ausführungsverhalten und für typische, gefährliche Fehlerquellen in Anwendungsprogrammen erlangt.
13. Inhalt:	Compilerarchitekturen im Überblick; lexikalische und syntaktische Analyse von Texten mit formaler Grammatik, insb. von Programmiersprachen. Lexikalische Analyse: endliche Automaten und ihre Implementierung; Syntaxanalyse: diverse Parser-Strategien, ihre Implementierung und Eigenschaften. Methoden der automatischen Generierung von Analysatoren aus Spezifikationen der Grammatiken. Fehlererkennung und -behandlung. Analyse der statischen Semantik: Grundbegriffe und elementare Methoden. Eigenschaften von Programmiersprachen; Realisierung der Laufzeitsemantik prozeduraler Programmiersprachen aus Benutzersicht, insbesondere Implementierungsmodelle der Speicherverwaltung und der Unterprogrammaufrufe. Vermeidung typischer Fehlerquellen und überraschender Probleme in Anwendungsprogrammen.
14. Literatur:	• Aho, Sethi, Ullman, Compilers - Principles, Techniques, and Tools, 1988 • Wilhelm, Maurer, Uebersetzerbau, 1997
15. Lehrveranstaltungen und -formen:	• 101501 Vorlesung Grundlagen des Compilerbaus und der Programmiersprachen • 101502 Übung Grundlagen des Compilerbaus und der Programmiersprachen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name:	10151 Grundlagen des Compilerbaus und der Programmiersprachen (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog SWT Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester → Ergänzendes Modul Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester → Wahlmodule

Modul: 10170 Imaging Science

2. Modulkürzel:	051900210	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch

8. Modulverantwortlicher:	Gunther Heidemann
9. Dozenten:	• Thomas Ertl • Daniel Weiskopf • Gunther Heidemann
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	• 080300100 Mathematik für Informatiker und Softwaretechniker
12. Lernziele:	Der Student / die Studentin beherrscht die Grundlagen der Repräsentation und Verarbeitung digitaler Bilder, kann Probleme aus dem Fachgebiet einordnen und selbständig mit den erlernten Algorithmen und Verfahren lösen. The student knows the basics of digital image representation and processing and is able to solve problems of the field using the methods presented in the course.
13. Inhalt:	• Grundlagen aus der Optik:Lochkamera, Linsengleichung • Bildaufnahme:Kameras, Objektive, Beleuchtung, Aufnahmeprozess • Bildrepräsentation:Diskretisierung, Farbräume, Bildformate (z.B. ppm, gif, jpeg) • Elementare Bildbearbeitung:Punktoperationen (z.B. Kontrastverstärkung, Binarisierung) • Lineare und nichtlineare Filter:Faltung, morphologische Operatoren • Fouriertransformation, Bilddarstellung und -bearbeitung im Fourierraum, Abtasttheorem • Orthogonale Transformationen:Cosinus, Wavelets • Kompression:Generische Verfahren (RLE, Entropie), spezielle Bildverfahren (z.B. jpeg) • Video:Formate, Kompression (z.B. AVI, MPEG) • Bildverbesserung und Restauration • Elementare Segmentierungsverfahren:Histogramme, Farben, Konturen • Elementare Mustererkennung • Fundamentals of optics such as pinhole camera and lens equation • Image acquisition: Cameras, lenses, illumination, acquisition process • Image representation: Discretization, color spaces, image file formats (e.g. ppm, gif, jpeg) • Basics of image processing, e.g. point operations such as contrast enhancement or binarization • Linear and nonlinear filtering such as convolution and morphological operations.

	• Fourier transform, image representation and processing in Fourier space, sampling theorem • Orthogonal transforms such as cosine transform and wavelets • Compression: Generic compression (RLE, entropy coding), methods specialized to domain of images (e.g. jpeg) • Video: file formats, compression (e.g. avi, mpeg) • Image enhancement and restauration • Basics of segmentation: Histograms, colors, contours • Basics of pattern recognition
14. Literatur:	• Bässmann, Henning; Kreyss, Jutta, Bildverarbeitung Ad Oculos, 2004 • Forsyth, David and Ponce, Jean, Computer Vision. A Modern Approach.: A Modern Approach Computer Vision. A Modern Approach, 2003 • Gonzalez, Rafael C.; Woods, Richard E.; Eddins, Steven L., Digital Image Processing, 2004 • Jähne, Bernd, Digitale Bildverarbeitung / Digital Image Processing, 2005 • Klaus D. Tönnies, Grundlagen der Bildverarbeitung, 2005 • L. G. Shapiro, G. C. Stockman, Computer Vision, 2001
15. Lehrveranstaltungen und -formen:	• 101701 Vorlesung Imaging Science • 101702 Übung Imaging Science
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10171 Imaging Science (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW

Modul: 10250 Parallele Systeme

2. Modulkürzel:	051200065	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	Ein Fach aus dem Bereich Technische Informatik.
12. Lernziele:	Grundlegende Kenntnisse im Bereich paralleler Systeme, z.B. Multi-Core CPUs und deren Programmierung.
13. Inhalt:	• Die Entwicklung vom klassichen Mikroprozessor zur Multi-Core CPU • Programmierung paralleler Rechnersysteme • Systolische Arrays, massiv parallele Systeme • Parallele Systeme aus verschiedenen • Anwendungsdomänen: ausgewählte Fallbeispiele
14. Literatur:	• Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	• 102501 Vorlesung Parallele Systeme • 102502 Übung Parallele Systeme
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10251 Parallele Systeme (LBP), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Technische Kybernetik, 3. Semester → Ergänzungsmodule → Höhere Informatik B.Sc. Softwaretechnik, 3. Semester → Ergänzungsmodule → Katalog ISW

Modul: 11110 Verteilte Systeme

2. Modulkürzel:	051200015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Kurt Rothermel
9. Dozenten:	Kurt Rothermel
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	• 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen • 051200005 Systemkonzepte und -Programmierung
12. Lernziele:	• Verstehen der grundätzlichen Eigenschaften, Konzepte und Verfahren verteilter Systeme. • Kann existierende verteilte Anwedungen und Systemplattformen hinsichtlich ihrer Eigenschaften analysieren und verstehen. • Kann verteilte Anwendungen/Systemplattformen auf der Grundlage der erlernten Methoden realisieren. • Kann sich mit Experten anderer Fachdiszipinen über die Anwendung verteilter Systeme verständigen.
13. Inhalt:	• Einführung in die verteilten Systeme • Systemmodelle • Kommunikation: Nachrichten, Remote Procedure Call (RPC), Remote Method Invocation RMI) • Namensgebung: Generierung und Resolution • Zeit und Uhren in verteilten Systemen: Anwendungen, logische Uhren, physikalische Uhren, Uhrensynchronisation • Prozesssynchronisation: Wechselseitiger Ausschluß • Globaler Zustand: Konzepte, Snapshot Algorithmus, verteiltes Debugging • Transaktionsmanagement: Serialisierbarkeit, Sperrverfahren, 2-Phasen-Commit-Protokolle • Datenreplikation: Primary Copy, Consensus-Protokolle und andere Algorithmen • Sicherheit: Verfahren zur Geheimhaltung, Integrität, Authentifikation und Autorisierung • Broadcast-Algorithmen: Verarbeitungsmodell, Broadcast-Semantiken und -Algorithmen
14. Literatur:	Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	• 111101 Vorlesung Verteilte Systeme • 111102 Übungen Verteilte Systeme
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 h Selbststudiumszeit / Nachbearbeitungszeit: 138 h Gesamt: 180 h

17. Prüfungsnummer/n und -name:	• 11111 Verteilte Systeme (PL), schriftliche Prüfung, 90 Min., Gewichtung: 7.0, • 11112 Verteilte Systeme - Übungen (LBP), schriftlich, eventuell mündlich, Gewichtung: 3.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Parallele und Verteilte Systeme
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Wirtschaftsinformatik, 5. Semester → Informatik (B 1) B.Sc. Wirtschaftsinformatik, 5. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich II: Informatik Basis B.Sc. Wirtschaftsinformatik, 5. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich III: Informatik Simulation

330 Katalog ISW 1-3

Zugeordnete Module:

10050	Bildverstehen
10060	Computergraphik
11900	Design and Test of Systems on a Chip
10170	Imaging Science
10180	Information Retrieval und Text Mining
10250	Parallele Systeme
10300	Rechnernetze

Modul: 10050 Bildverstehen

2. Modulkürzel:	051200035	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Paul Levi
9. Dozenten:	• Paul Levi • Viktor Avrutin
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISW
11. Voraussetzungen:	Grundwissen über Programmierung, Datenstrukturen und Mathematik.
12. Lernziele:	Die Teilnehmer kennen und verstehen die Grundlagen der klassischen, verkörperten und verteilten Künstlichen Intelligenz. Ihnen wurden die dabei verwendeten Grundbegriffe so vorgestellt, dass sie als Bausteine von Architekturen intelligenter Systeme betrachtet werden. Dies bedeutet im Einzelnen, dass die Zuhörer am Beispiel des Bildverstehens die Wirkungsweise kognitiver Wahrnehmungsfähigkeiten kennengelernt haben, verstehen wie diese Fähigkeiten im Rahmen des Beobachtungs-, Planungs- und Aktionszyklus in einem Agenten eingesetzt werden und wie diese Fähigkeiten erweitert werden müssen, damit einzelne Agenten sich in einem Team kooperativ (Multiagenten-Systeme) verhalten können. Die Studierenden sind in der Lage, nicht nur einzelne wesentliche Algorithmen der Künstlichen Intelligenz (einschließlich des Bildverstehens) zu verstehen, sondern auch die wechselseitigen Beziehungen von verschiedenen Methoden zu berücksichtigen, um eine Beurteilung der Konzeption und der Wirkungsweise von intelligenten (kognitiven) technischen Systemen selbst vornehmen zu können.
13. Inhalt:	• Einleitung in Problemstellungen und Definitionen • Einführung in die Grundlagen der Bildverarbeitung • (Künstliche) Neuronale Netze • Bedingungsausbreitung (Constraints und ihre Propagierung) • Probabilistische Inferenzmit Bayes-Netzwerke • Verteilte Künstliche Intelligenz (VKI) • Multiagentensysteme (MAS)
14. Literatur:	• G. F. Luger, Künstliche Intelligenz, 2001 • Jähne, Bernd, Digitale Bildverarbeitung, 2005 • Literatur, siehe Webseite zur Veranstaltung • S. Russell, P. Norvig, Künstliche Intelligenz, 2004
15. Lehrveranstaltungen und -formen:	• 100501 Vorlesung Bildverstehen • 100502 Übung Bildverstehen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name:	10051 Bildverstehen (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Technische Kybernetik, 4. Semester → Ergänzungsmodule → Grundlagen der Natur- und Ingenieurwissenschaften B.Sc. Technische Kybernetik, 4. Semester → Ergänzungsmodule → Höhere Informatik B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW

Modul: 10060 Computergraphik

2. Modulkürzel:	051900002	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Daniel Weiskopf
9. Dozenten:	• Thomas Ertl • Daniel Weiskopf
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISW
11. Voraussetzungen:	• Modul 051900001 Mensch-Computer-Interaktion • Modul 051240005 Numerik und Stochastik.
12. Lernziele:	Die Studierenden haben Wissen über die Grundlagen der Computergraphik sowie praktische Fähigkeiten in der Graphikprogrammierung erworben.
13. Inhalt:	Folgende Themen werden in der Vorlesung behandelt: • Überblick über den Bildsyntheseprozess • Grundlegende Rastergraphik • 2D und 3D Geometrietransformationen, 3D Projektion • Polygonale und hierarchische Modelle • Verdeckungsberechung • Grundlegende Renderingtechniken (Rasterung, Raytracing) • Beleuchtungsmodelle • Texturen • Grundlagen der geometrischen Modellierung (Kurven, Flächen) Die Veranstaltung besteht aus Vorlesung mit Übungen. Die Übungen umfassen praktische Programmierübungen, theoretische Themen und Programmierprojekte.
14. Literatur:	• J. Encarnacao, W. Strasser, R. Klein, Graphische Datenverarbeitung (Band1 und 2), 1997 • J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics: Principle and Practice, 1990
15. Lehrveranstaltungen und -formen:	• 100601 Vorlesung Computergraphik • 100602 Übung Computergraphik
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10061 Computergraphik (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Maschinelle Sprachverarbeitung, 5. Semester → Wahlbereich E/I

Modul: 11900 Design and Test of Systems on a Chip

2. Modulkürzel:	051700015	5. Moduldauer:	-
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch

8. Modulverantwortlicher:	Hans-Joachim Wunderlich
9. Dozenten:	• Hans-Joachim Wunderlich • Melanie Elm
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISW
11. Voraussetzungen:	• Modul 051700005 Rechnerorganisation • Modul 051700010 Grundlagen der Rechnerarchitektur
12. Lernziele:	The students of this course have gained a basic understanding of development and test of complex embedded hardware / software systems. The participants have become acquainted with the essential steps of synthesis, validation, test and programming and have learned, how to use the related tools for automation. Besides the different design styles, paradigms and standards the essential steps of automated design, test and programming of digital and mixed signal circuits have been discussed. Exercises and labs have served to practice the use of commercial tools and designs.
13. Inhalt:	The course comprises: • Overview of system design • Reuse and cores • Standards and platforms • Elements of analog and mixed signal designs • Design validation and verification • Test and design for testability with the related standards • Application and programming of embedded processors
14. Literatur:	• A. Sloss, D. Symes, C. Wright, ARM System Developer's Guide: Designing and Optimizing System Software, 2004 • L.-T. Wang, C.-W. Wu, X. Wen, VLSI Test Principles and Architectures - Design for Testability, 2006 • M. Keating, P. Bricaud, Reuse Methodology Manual for System-on-a-Chip Designs, 2007 • M. L. Bushnell, V. D. Agrawal, Essentials of Electronic Testing, 2005 • S. Furber, ARM System-on-Chip Architecture, 2000 • W. Wolf, Modern VLSI Design: System-on-Chip Design, 2002
15. Lehrveranstaltungen und -formen:	• 119001 Vorlesung Design and Test of Systems on a Chip • 119002 Übung Design and Test of Systems on a Chip • 119003 Praktikum Design and Test of Systems on a Chip
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name:	11901	Design and Test of Systems on a Chip (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:		
21. Zuordnung zu weiteren Curricula:		

Modul: 10170 Imaging Science

2. Modulkürzel:	051900210	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch

8. Modulverantwortlicher:	Gunther Heidemann
9. Dozenten:	• Thomas Ertl • Daniel Weiskopf • Gunther Heidemann
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	• 080300100 Mathematik für Informatiker und Softwaretechniker
12. Lernziele:	Der Student / die Studentin beherrscht die Grundlagen der Repräsentation und Verarbeitung digitaler Bilder, kann Probleme aus dem Fachgebiet einordnen und selbständig mit den erlernten Algorithmen und Verfahren lösen. The student knows the basics of digital image representation and processing and is able to solve problems of the field using the methods presented in the course.
13. Inhalt:	• Grundlagen aus der Optik:Lochkamera, Linsengleichung • Bildaufnahme:Kameras, Objektive, Beleuchtung, Aufnahmeprozess • Bildrepräsentation:Diskretisierung, Farbräume, Bildformate (z.B. ppm, gif, jpeg) • Elementare Bildbearbeitung:Punktoperationen (z.B. Kontrastverstärkung, Binarisierung) • Lineare und nichtlineare Filter:Faltung, morphologische Operatoren • Fouriertransformation, Bilddarstellung und -bearbeitung im Fourierraum, Abtasttheorem • Orthogonale Transformationen:Cosinus, Wavelets • Kompression:Generische Verfahren (RLE, Entropie), spezielle Bildverfahren (z.B. jpeg) • Video:Formate, Kompression (z.B. AVI, MPEG) • Bildverbesserung und Restauration • Elementare Segmentierungsverfahren:Histogramme, Farben, Konturen • Elementare Mustererkennung • Fundamentals of optics such as pinhole camera and lens equation • Image acquisition: Cameras, lenses, illumination, acquisition process • Image representation: Discretization, color spaces, image file formats (e.g. ppm, gif, jpeg) • Basics of image processing, e.g. point operations such as contrast enhancement or binarization • Linear and nonlinear filtering such as convolution and morphological operations.

	• Fourier transform, image representation and processing in Fourier space, sampling theorem • Orthogonal transforms such as cosine transform and wavelets • Compression: Generic compression (RLE, entropy coding), methods specialized to domain of images (e.g. jpeg) • Video: file formats, compression (e.g. avi, mpeg) • Image enhancement and restauration • Basics of segmentation: Histograms, colors, contours • Basics of pattern recognition
14. Literatur:	• Bässmann, Henning; Kreyss, Jutta, Bildverarbeitung Ad Oculos, 2004 • Forsyth, David and Ponce, Jean, Computer Vision. A Modern Approach.: A Modern Approach Computer Vision. A Modern Approach, 2003 • Gonzalez, Rafael C.; Woods, Richard E.; Eddins, Steven L., Digital Image Processing, 2004 • Jähne, Bernd, Digitale Bildverarbeitung / Digital Image Processing, 2005 • Klaus D. Tönnies, Grundlagen der Bildverarbeitung, 2005 • L. G. Shapiro, G. C. Stockman, Computer Vision, 2001
15. Lehrveranstaltungen und -formen:	• 101701 Vorlesung Imaging Science • 101702 Übung Imaging Science
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10171 Imaging Science (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW

Modul: 10180 Information Retrieval und Text Mining

2. Modulkürzel:	052401010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Hinrich Schütze
9. Dozenten:	• Helmut Schmid • Hinrich Schütze
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISW
11. Voraussetzungen:	052400009
12. Lernziele:	Die Studierenden haben ein grundlegendes Verständnis der Konzepte und Algorithmen des Information Retrieval und Text Mining entwickelt.
13. Inhalt:	• Textpräprozessierung • invertierte Indexe • IR-Modelle (z.B. Vektorraum-basiertes IR) • Linkanalyse • Clustering • Frage-Antwort-Systeme • Informationsextraktion • korpusbasierter Erwerb von lexikalischem und Weltwissen
14. Literatur:	• Chris Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, 2008 Cambridge University Press
15. Lehrveranstaltungen und -formen:	• 101801 Vorlesung Information Retrieval and Text Mining • 101802 Übung Information Retrieval and Text Mining
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden
17. Prüfungsnummer/n und -name:	10181 Information Retrieval und Text Mining (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Maschinelle Sprachverarbeitung, 5. Semester → Kernmodule

Modul: 10250 Parallele Systeme

2. Modulkürzel:	051200065	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG
11. Voraussetzungen:	Ein Fach aus dem Bereich Technische Informatik.
12. Lernziele:	Grundlegende Kenntnisse im Bereich paralleler Systeme, z.B. Multi-Core CPUs und deren Programmierung.
13. Inhalt:	• Die Entwicklung vom klassichen Mikroprozessor zur Multi-Core CPU • Programmierung paralleler Rechnersysteme • Systolische Arrays, massiv parallele Systeme • Parallele Systeme aus verschiedenen • Anwendungsdomänen: ausgewählte Fallbeispiele
14. Literatur:	• Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	• 102501 Vorlesung Parallele Systeme • 102502 Übung Parallele Systeme
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10251 Parallele Systeme (LBP), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Technische Kybernetik, 3. Semester → Ergänzungsmodule → Höhere Informatik B.Sc. Softwaretechnik, 3. Semester → Ergänzungsmodule → Katalog ISW

Modul: 10300 Rechnernetze

2. Modulkürzel:	051200010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Kurt Rothermel
9. Dozenten:	• Kurt Rothermel • Frank Dürr
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISW
11. Voraussetzungen:	• 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen
12. Lernziele:	• Versteht grundsätzliche Eigenschaften, Konzepte und Methoden von Rechnernetzen, inbesondere dem Internet. • Versteht Schichten und deren Zusammenwirken in einem Protokollstapel • Kann Rechnernetze aufbauen, verwalten und analysieren. • Kann Protokolle entwickeln und in Schichtenarchitektur einbetten. • Kann höhere Kommunikationsdienste zur Entwicklung von netzgestützen Systemen anwenden. • Kann sich mit Experten anderer Domänen über Methoden der Rechnernetze verständigen.
13. Inhalt:	• Einführung in die Rechnernetze, ISO Referenzmodell; • Bitübertragungsschicht: Übertragungsmedien, analoge und digitale Informationskodierung und -übertragung, Vermittlungsarten; • Sicherungsschicht: Betriebsarten, Fehlererkennung und -behandlung, Flusskontrolle; • Lokale Netze: CSMA/CD, Token Ring, Token Bus, FDDI, Kopplung; • Vermittlungsschicht: Verbindungsorientierter und verbindungsloser Dienst, Leitwegbestimmung, Überlastkontrolle; • Internetworking; • Internet-Protokoll; • Transportschicht: ausgewählte Realisierungsprobleme und Internet-Protokolle; • Echtzeitkommunikation: IntServ, DiffServ; Sicherheit: Verfahren, IPsec, SSL, TLS.
14. Literatur:	• A.S. Tanenbaum, Computer Networks, 4th Edition, 2003 • D.E. Comer, Computernetzwerke und Internets, 2000 • D.E. Comer, Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture, 1995 • J. F. Kurose, K. W. Ross, Computer Networks: a top-down approach featuring the Internet, 2001 • L.L. Peterson, B.S. Davie, Computer Networks: A Systems Approach, 1999
15. Lehrveranstaltungen und -formen:	• 103001 VL Rechnernetze • 103002 ÜB Rechnernetze

16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	• 10301 Rechnernetze (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 7.0 • 10302 Rechnernetze - Übungen (LBP), schriftlich, eventuell mündlich, Gewichtung: 3.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Technische Kybernetik, 4. Semester → Ergänzungsmodule → Höhere Informatik B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW

340 Katalog ISW 4-7

Zugeordnete Module: 10160 Hardwarebeschreibungssprachen

Modul: 10160 Hardwarebeschreibungssprachen

2. Modulkürzel:	051711015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	2.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Martin Radetzki
9. Dozenten:	• Martin Radetzki • Sven Simon
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Katalog ISW 4-7 BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISW
11. Voraussetzungen:	• Modul 051711005 Technische Grundlagen der Informatik oder • Modul Einführung in die Technische Informatik (aus Studiengang Softwaretechnik, Modulnummer wird nachgetragen)
12. Lernziele:	• Kenntnis Hardwarebeschreibungssprachen • Kenntnis VHDL
13. Inhalt:	Hardwarebeschreibungssprachen werden eingesetzt, um digitale Schaltungen auf der Algorithmen-, Registertransfer- und Gatterebene zu beschreiben. Sie ermöglichen eine ereigenisgesteuerte Simulation und stellen Konzepte zur Beschreibung von Hierarchie, Nebenläufigkeit und Zeit zusätzlich zu den von Programmiersprachen bekannten Eigenschaften zur Verfügung. Dieses Modul vermittelt die wichtigsten Sprachkonstrukte der Hardwarebeschreibungssprache VHDL und ihre sinnvolle Anwendung zur Beschreibung digitaler Schaltungen und Systeme, indem die folgenden Themen vorgestellt und in Rechnerübungen angewandt werden: • Entwurfshierarchie: Entities, Architectures, Instanzen, Verbindungen • Quellcodedateien und ihre Kompilation in Entwurfsbibliotheken • Nebenläufige sequentielle Prozesse • Sequentielle Anweisungen in VHDL • Typsystem von VHDL • Beschreibung typischer Hardwarestrukturen • VHDL für die Hardwaresynthese, Synthesesemantik • Beschreibung regulärer und rekursiver Strukturen • Simulationsmechanismen • Testbenches and Handhabung von Dateien • Organisation VHDL-basierter Projekte
14. Literatur:	• P. J. Ashenden, The Designer's Guide to VHDL (2nd edition), 2002 • P. J. Ashenden, The Student's Guide to VHDL, 1998
15. Lehrveranstaltungen und -formen:	• 101601 Vorlesung Hardwarebeschreibungssprachen • 101602 Übung Hardwarebeschreibungssprachen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 21 Stunden

	Nachbearbeitungszeit: 69 Stunden
17. Prüfungsnummer/n und -name:	• 10161 Hardwarebeschreibungssprachen (PL), schriftlich, eventuell mündlich, 60 Min., Gewichtung: 1.0 • 10162 Hardwarebeschreibungssprachen - Rechnerübungen (LBP), schriftlich, eventuell mündlich, Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	

350 Wahlmodule aus Master Informatik

Zugeordnete Module:

29560	Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems
10040	Bildsynthese
29570	Computer Interface Technologien
29580	Datenkompression
29590	Digitaler Systementwurf I
29440	Geometrische Modellierung und Animation
29610	Hardware Based Fault Tolerance
14380	Hardware Verification and Quality Assessment
29640	Mikrocontroller
29500	Visual Computing
11330	Visualisierung

Modul: 29560 Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems

2. Modulkürzel:	051700024	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch

8. Modulverantwortlicher:	Hans-Joachim Wunderlich
9. Dozenten:	• Melanie Elm • Hans-Joachim Wunderlich
10. Zuordnung zum Curriculum:	B.Sc. Informatik → Ergänzungsmodule → Wahlmodule aus Master Informatik
11. Voraussetzungen:	10310 Rechnerorganisation oder 10140 Grundlagen der Rechnerarchitektur / Advanced Processor Architecture
12. Lernziele:	*Knowledge of the most important algorithms and methods in design automation tools at any design level*
13. Inhalt:	Firstly, the lecture points out the basic algorithms in modern design automation software. Next, the problems occurring in synthesis, analysis and test of digital circuits at the different design levels are discussed and their solutions are mapped to the basic algorithms. Major aspects in the discussion are the challenges and problems arising from nanometer technology. Here the focus always lies on the software supporting the design of digital systems.
14. Literatur:	G. De Micheli: Synthesis and Optimization of Digital Circuits, McGrawHill, New York, NY, USA, 1994. Hachtel, G. D. and Somenzi: Logic Synthesis and Verification Algorithms, Kluwer Academic Publishers, Norwell, MA, USA, 2000. Ban Wong, Anurag Mittal, Yu Cao: Nano-CMOS Circuit and Physical Design, John Wiley & Sons Inc, 2004. Ashish Srivastava, Dennis Sylvester, David Blaauw: Statistical Analysis and Optimization for VLSI: Timing and Power, Springer, 2005. Jens Lienig: Layoutsynthese elektronischer Schaltungen - Grundlegende Algorithmen fuer die Entwurfsautomatisierung, Springer, 2006. L.-T. Wang, Y.-W. Chang, K.-W. Cheng: Electronic Design Automation, Morgan Kaufmann, 2009
15. Lehrveranstaltungen und -formen:	• 295601 Vorlesung Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems • 295602 Übung Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems
16. Abschätzung Arbeitsaufwand:	**Präsenzzeit:** **42 Stunden** **Selbststudium:** **108 Stunden** **Summe:** **150 Stunden**

17. Prüfungsnummer/n und -name:	29561	Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Klausur 90 min. oder mündliche Prüfung 30 min.
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:	Institut für Technische Informatik	
21. Zuordnung zu weiteren Curricula:		

Modul: 10040 Bildsynthese

2. Modulkürzel:	051900012	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Thomas Ertl
9. Dozenten:	• Martin Fuchs • Thomas Ertl • Daniel Weiskopf
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Wahlmodule aus Master Informatik BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISW
11. Voraussetzungen:	• Modul 051900002 Computergraphik • Modul 051900001 Mensch-Computer-Interaktion
12. Lernziele:	Die Studierenden besitzen Wissen über verschiedene Ansätze und Algorithmen der dreidimensionalen Computergraphik, physikalisch-basierte Verfahren wie Raytracing und Radiosity, die den Lichttransport und die Wechselwirkung mit Materie modellieren, und numerische Methoden wie Monte-Carlo-Integration und Finite-Elemente-Verfahren die es erlauben, die Rendering-Gleichung zu lösen. Interaktive Verfahren nutzen spezielle Eigenschaften moderner Graphikhardware, um mit Hilfe mehrdimensionaler Texturen und anderer Rasterisierungsoperationen realistische Beleuchtungseffekte in Echtzeit zu generieren. Bild-basierte Verfahren verzichten auf eine geometrische Repräsentation der Szene und erzeugen neue Ansichten aus anderen aufgenommenen Bildern.
13. Inhalt:	In dieser Vorlesung werden die folgenden Themen behandelt: • Grafik Hardware und APIs, OpenGL • Texturen, prozedurale Modelle • Schattenberechnungen • Szenengraphen, Culling, Level-of-Detail Verfahren • Physikalisch-basierte Beleuchtungsberechnung, Fotorealistische Bildsynthese • Lokale Beleuchtungsmodelle • Raytracing, Monte-Carlo Methoden • Radiosity
14. Literatur:	• Andrew S. Glassner, Principles of Digital Image Synthesis, 1995 • D. Eberly, 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, 2000 • J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics: Principle and Practice, 1990 • Literatur, siehe Webseite zur Veranstaltung • P. Dutre, P. Bekaert, K. Bala, Advanced Global Illumination, 2003 • Tomas Akenine-Möller, Eric Haines, Real-Time Rendering, 2002

	• Matt Pharr, Greg Humphreys, Physically Based Rendering: From Theory To Implementation, Morgan Kaufmann; Auflage: 2nd revised edition. (26. August 2010) • Peter Shirley et al, Fundamentals of Computer Graphics, Third Edition, A.K. Peters, July 2009
15. Lehrveranstaltungen und -formen:	• 100401 Vorlesung Bildsynthese • 100402 Übung Bildsynthese
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10041 Bildsynthese (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISW

Modul: 29570 Computer Interface Technologien

2. Modulkürzel:	051230105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch

8. Modulverantwortlicher:	Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum:	B.Sc. Informatik → Ergänzungsmodule → Wahlmodule aus Master Informatik
11. Voraussetzungen:	Kenntnisse in mindestens einem Fach der Technichen Informatik oder ähnlichen Fächern und Erfahrungen in mindestens einer Programmiersprache.
12. Lernziele:	Die Studierenden haben die Funktionsweise und den Aufbau von Coputer Interfaces verstanden. Sie beherrschen verschiedene Interface-Konzepte und kennen die Eigenschaften der Datenströme wie Latenzzeit, tatsächliche Durchsatzrate, Echtzeitfähigkeit, Umgang mit Übertragungsfehlern, etc.
13. Inhalt:	• Grundlagen - Computer Interfaces • Computer Interfaces und OSI-Modelle • Bus- und Netz-Topologien • Line und Error Codes • Protokolle • Treiber • Compliance Tests • Standardization Groups: USB, PCI, etc.
14. Literatur:	• Patterson, David A.; Hennessey, John L., Computer Organization and Design - The Hardware / Software Interface, 2008 More literature is named in the lecture.
15. Lehrveranstaltungen und -formen:	295701 Vorlesung mit Übung Computer Interface Technologien
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden
17. Prüfungsnummer/n und -name:	29571 Computer Interface Technologien (PL), schriftlich oder mündlich, Gewichtung: 1.0, Schriftliche Prüfung von 120 Minuten oder mündliche Prüfung von 30 Min.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	

Modul: 29580 Datenkompression

2. Modulkürzel:	051230110	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankündigung

8. Modulverantwortlicher:	Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum:	B.Sc. Informatik → Ergänzungsmodule → Wahlmodule aus Master Informatik
11. Voraussetzungen:	Grundlagenkenntnisse im Fach Mathematik.
12. Lernziele:	Die Studierenden haben die Konzepte der Datenkompression erlernt und haben Verständnis über verschiedene Algorithmen zur Datenkompression erworben. Sie können die behandelten Algorithmen implementieren und weiterentwickeln.
13. Inhalt:	• Shannons Entropie-Definition • Huffman Codierung • Arithmetische Codierung • Universal Codes • Verlustlose u. verlustbehaftete Kompression • Bilddatenkompression • Wörterbuch-basierte Kompression
14. Literatur:	• Khalid Sayood, Introduction to Data Compression, 2005 • More literature is named in the lecture
15. Lehrveranstaltungen und -formen:	295801 Vorlesung mit Übung Datenkompression
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden
17. Prüfungsnummer/n und -name:	29581 Datenkompression (PL), schriftlich oder mündlich, Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündliche Prüfung von 30 Min.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	

Modul: 29590 Digitaler Systementwurf I

2. Modulkürzel:	051230120	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch

8. Modulverantwortlicher:	Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum:	B.Sc. Informatik → Ergänzungsmodule → Wahlmodule aus Master Informatik
11. Voraussetzungen:	Kenntnisse in einem Fach aus der Technischen Informatik oder einem ähnlichen Gebiet.
12. Lernziele:	Die Studierende beherrschen den Entwurf Digitaler Systeme durch die Integration von digitalen Komponenten auf einem Boad und die Realisierung von digitaler Komponenten mittels FPGAs.
13. Inhalt:	• Praktische Einführung in den System-Entwurf mit digitalen Komponenten wie Schnittstellenbausteinen zur Kommunikation, FPGAs, Prozessoren, intelligenten Sensoren etc. • Einführung und Verwendung der Hardware-Beschreibungssprache VHDL zum Entwurf Digitaler Systeme • Digitale Systeme und Board-Integration von digitalen Komponenten • Aufbau von Computer-Boards u. Gbit/s-Interconnects • Entwurf auf höheren Abstraktionsebenen zur schnellen Entwicklung von Prototypen
14. Literatur:	• Kou-Chuan Chang, K.C. Chang, Digital Systems Design with VHDL and Synthesis: An Integrated Approach, 1999 More literature is named in the lecture.
15. Lehrveranstaltungen und -formen:	
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden
17. Prüfungsnummer/n und -name:	29591 Digitaler Systementwurf I (PL), schriftlich oder mündlich, Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündliche Prüfung von 30 Min.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	

Modul: 29440 Geometrische Modellierung und Animation

2. Modulkürzel:	051900010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch

8. Modulverantwortlicher:	Daniel Weiskopf
9. Dozenten:	• Thomas Ertl • Daniel Weiskopf • Carsten Dachsbacher
10. Zuordnung zum Curriculum:	B.Sc. Informatik → Ergänzungsmodule → Wahlmodule aus Master Informatik
11. Voraussetzungen:	Modul 051900002 Computergraphik Modul 051900001 Mensch-Computer-Interaktion Modul 051240005 Numerik und Stochastik
12. Lernziele:	Die Studierende kennen die theoretischen Grundlagen der geometrischen Modellierung und Animation und haben praktische Fähigkeiten in der Programmierung von Verfahren der geometrischen Modellierung und Animation sowie in der Benutzung von Modellierungs- und Animationssoftware wie Maya erlangt.
13. Inhalt:	Die Vorlesung behandelt Grundlagen und ausgewählte Verfahren der Szenenmodellisierung und der Computeranimation. Dazu gehören Kenntnisse über Kurven- und Flächenrepräsentationen, wie sie in Animationspaketen zur Modellierung von Objekten, zur Beschreibung von dynamischen Verhalten von Parametern und zur Keyframe-Interpolation verwendet werden. Physikalisch-basierte Animation hingegen beschreibt Bewegung durch die kinematischen und dynamischen Gesetze der Mechanik. Anwendungen reichen von Partikelsystemen bis zur Simulation von mehrgliedrigen Modellen und Verformungen. Folgende Themen werde in der Vorlesung behandelt: • Kurvenbeschreibung und -modellisierung (allg. Polynomkurven, Bezier-Kurven, B-Slines, NURBS) • Flächenmodellisierung (Tensorproduktflächen, NURBS) • Unterteilungsschemata • Überblick über Animationstechniken • Keyframe-Animation • Physikalisch basierte Animation Die Veranstaltung besteht aus Vorlesung mit Übungen. Die Übungen umfassen praktische Proframmierübungen und Anwendungen von Modellierungs- und Animationswerkzeugen (wie Maya) sowie theoretische Themen.
14. Literatur:	• D. Eberly, 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, 2000

	• G. Farin, Curves ans Surfaces for CAGD: A Practical Guide, 2002 • R. Parent, Computer Animation: Algorithms and Techniques, 2002
15. Lehrveranstaltungen und -formen:	294401 Vorlesung mit Übungen Geometrische Modellierung und Animation
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden
17. Prüfungsnummer/n und -name:	29441 Geometrische Modellierung und Animation (PL), schriftlich oder mündlich, Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündlichen Prüfung von 30 Min. Written exam of 120 min. or oral exam of 30 min.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	

Modul: 29610 Hardware Based Fault Tolerance

2. Modulkürzel:	051710023	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch

8. Modulverantwortlicher:	Hans-Joachim Wunderlich
9. Dozenten:	• Hans-Joachim Wunderlich • Michael Kochte • Melanie Elm
10. Zuordnung zum Curriculum:	B.Sc. Informatik → Ergänzungsmodule → Wahlmodule aus Master Informatik
11. Voraussetzungen:	10140 Grundlagen der Rechnerarchitektur / Advanced Processor Architecture 10310 Rechnerorganisation
12. Lernziele:	Knowledge of methods for reliability assessment of circuits and systems Knowledge of error control coding Knowledge how to design self checking circuits
13. Inhalt:	Micro- and Nano-electronic systems can exhibit failures both right after production and during their operation. Systems for which safety and security is of concern have to be designed in a way that the desired function can be delivered even if some components fail or produce erroneous outputs. This lecture presents the most important design techniques that allow to tolerate hardware faults up to a certain degree. The topics of the lecture are as follows: Terminology Measures of fault tolerance Techniques for structural and time redundancy Error detection and diagnosis Fault masking, repair, reconfiguration Fault-tolerant distributed systems
14. Literatur:	Apart from lecture slides, the following books can be used to deepen on the topics of the lecture: I. Koren and C. M. Krishna, Fault-Tolerant Systems Morgan-Kaufman, 2007 P. K. Lala, Self-Checking and Fault-Tolerant Digital Design, Morgan Kaufmann Publishers (2001) D.K. Pradhan, Fault-Tolerant Computer Design, Prentice Hall (1996) R.N. Rao, E. Fujiwara, Error Control Coding for Computer Systems, Prentice Hall (1989) M.L. Bushnell, V.D. Agrawal, Essentials of Electronic Testing, Klumer Academic Publishers (2000)
15. Lehrveranstaltungen und -formen:	• 296101 Vorlesung Hardware Based Fault Tolerance • 296102 Übung Hardware Based Fault Tolerance
16. Abschätzung Arbeitsaufwand:	Presence Time: 42 Stunden

	Self Study: 138 Stunden Sum: 180 Stunden
17. Prüfungsnummer/n und -name:	29611 Hardware Based Fault Tolerance (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Written exam 90 min or Oral exam 30 min
18. Grundlage für ... :	
19. Medienform:	Laptop presentation
20. Angeboten von:	Institut für Technische Informatik
21. Zuordnung zu weiteren Curricula:	

Modul: 14380 Hardware Verification and Quality Assessment

2. Modulkürzel:	051700020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch

8. Modulverantwortlicher:	Hans-Joachim Wunderlich
9. Dozenten:	• Hans-Joachim Wunderlich • Michael Kochte
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Wahlmodule aus Master Informatik BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISW
11. Voraussetzungen:	• 051700005 Rechnerorganisation • 051711005 Technische Grundlagen der Informatik
12. Lernziele:	• Basic knowledge of methodologies and algorithms of functional and formal verification, diagnosis, test and design for testability of integrated circuits • Application of tools for simulation, verification and test insertion
13. Inhalt:	Complex integrated circuits and systems are hardly designed fault free at first go. Also during production defects and an imperfect yield have to be expected. The course deals with the basic techniques to find and locate faults and defects in the design and in the manufactured, integrated system. The discussed methods are applied with the help of commercial and academic tools in exercises and labs. The course comprises: • Validation: Simulation and emulation in different design levels. • Formal verification: Equivalence checking and model checking. • Test: Fault simulation and test generation. • Debug and diagnosis.
14. Literatur:	• G. D. Hachtel, F. Somenzi, Logic Synthesis and Verification Algorithms, 2006 • K. L. McMillan, Symbolic Model Checking, 1993 • L.-T. Wang, C.-W. Wu, X. Wen, VLSI Test Principles and Architectures - Design for Testability, 2006 • M. L. Bushnell, V. D. Agrawal, Essentials of Electronic Testing, 2005 • R. Drechsler, B. Becker, Graphenbasierte Funktionsdarstellung, 2000 • S. Hassoun, T. Sasao, Logic Synthesis and Verification, 2002 • S. Minato, Binary Decision Diagrams and Applications for VLSI CAD, 1996 • T. Kropf, Introduction to Formal Hardware Verification, 1999 • W. Kunz, D. Stoffel, Reasoning in Boolean Networks, 1997
15. Lehrveranstaltungen und -formen:	• 143801 Vorlesung Hardware Verification and Quality Assessment • 143802 Übung Hardware Verification and Quality Assessment
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name:	14381 Hardware Verification and Quality Assessment (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW

Modul: 29640 Mikrocontroller

2. Modulkürzel:	051230115	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch

8. Modulverantwortlicher:	Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum:	B.Sc. Informatik → Ergänzungsmodule → Wahlmodule aus Master Informatik
11. Voraussetzungen:	Kenntnisse in mindestens einer Programmiersprache und in mindestens einem Fach aus dem Bereich dem Bereich der Technischen Informatik oder ähnlichen Fächern.
12. Lernziele:	Studierende beherrschen die praktische Programmierung von Mikrokontrollern und kennen klassische Architekturen. • Historische Übersicht • Mikrocontroller-Architekturen • Einsatzgebiete von Mikrocontrollern • Befehlssatz klassischer Microcontroller • Assembler-Programmierung von Mikrocontrollern • C-Programmierung von Mikrocontrollern
13. Inhalt:	Als Microcontroller (auch ÂµController, ÂµC, MCU) werden ICs bezeichnet, die mit dem Prozessor mindestens Peripheriefunktionen auf einem Chip vereinen. In vielen Fällen befindet sich der Arbeits- und Programmierspeicher ebenfalls teilweise oder komplett auf dem gleichen Chip. Ein Mikrocontroller ist praktisch ein Ein-Chip-Computersystem. Die Anzahl der verbauten Mikrocontroller überschreitet bei weitem die Zahl der Mikropozessoren. Der Mikrocontroller tritt in Gestalt von eingebetteten Systemen im Alltag oft unbemerkt in technischen Gebrauchsartikeln auf, zum Beispiel in Waschmaschinen, Chipkarten (Geld-, Telefonkarten), Unterhaltungselektronik (Videorekordern, CD-/DVD-Playern, Radios, Fernsehgeräten, Fernbedienungen), Büroelektronik, Kraftfahrzeugen (Steuergeräte für z.B. ABS, Airbag, Motor, Kombiinstrument, ESP usw.), Mobiltelefonen und sogar in Uhren und Armbanduhren. Darüber hinaus sind sie in praktisch allen Computer-Peripheriegeräten enthalten (Tastatur, Maus, Drucker, Monitor, Scanner uvm.). Mikrocontroller sind in Leistung und Ausstattung auf die jeweilige Anwendung angepasst. Daher haben sie gegenüber normalen Computern deutliche Vorteile bei den Kosten und der Leistungsaufnahme. Kleine Mikrocontroller sind in höheren Stückzahlen für deutlich unter 1â, - verfügbar. Aus http://de.wikipedia.org/wiki/Mikrocontroller
14. Literatur:	• Jörg Wiegelmann, Softwareentwicklung in C für Mikroprozessoren und Mikrocontroller: C- Programmierung für Embedded-Systeme, 2009 More literature is named in the lecture

15. Lehrveranstaltungen und -formen:	296401 Vorlesung mit Übung Mikrocontroller
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden
17. Prüfungsnummer/n und -name:	29641 Mikrocontroller (PL), schriftlich oder mündlich, Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündlichen Prüfung von 30 Min.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	

Modul: 29500 Visual Computing

2. Modulkürzel:	051900014	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch

8. Modulverantwortlicher:	Daniel Weiskopf
9. Dozenten:	Carsten Dachsbacher
10. Zuordnung zum Curriculum:	B.Sc. Informatik → Ergänzungsmodule → Wahlmodule aus Master Informatik
11. Voraussetzungen:	Modul 051900002 Computergraphik Modul 051900001 Mensch-Computer-Interaktion
12. Lernziele:	Die Studierenden kennen die theoretischen Grundlagen des Visual Computing und haben praktische Fähigkeiten in der Programmierung von Techniken für Visual Computing erlangt. Die Studierenden kennen verschiedene Ansätze und Algorithmen, die für Visual Computing einsetzbar sind.
13. Inhalt:	Diese Vorlesung behandelt die digitale Verarbeitung visueller Information die sich auf Methoden der Computergrafik, algorithmischen Geometrie, Bildbearbeitung aber auch Computer Vision und maschinelles Lernen stützt. Die Vorlesung gliedert sich in fünf Themenbereiche: Grundlagen in Signalverarbeitung, Geometrie und Farblehre, Bildverarbeitung, Bildsynthese, Geometrierepräsentation und Computing Vision. In dieser Vorlesung werden die folgenden Themen behandelt: • Farbe und Licht • Abtastung, Sampling Theorem, Quantisierung • Transformationen, Kameraabbildungen, Projektionen • Bildtransformationen und -filterung, Auflösungspyramiden, Wavelets, Kompression • Segmentierung, Optischer Fluss • Grafik-Pipeline, Beleuchtungsberechnung, Texturen, Grafik-Hardware • Beschreibung von Kurven und Flächen • Maschinelles Lernen
14. Literatur:	• Andrew S. Glassner, Principles of Digital Image Synthesis, 1995 • J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics: Principle and Practice, 1990 • Jähne, Bernd, Digitale Bildverarbeitung, 2005 • Literatur, siehe Webseite zur Veranstaltung • M. Pharr, G. Humphreys, Physically Based Rendering, 2004
15. Lehrveranstaltungen und -formen:	295001 Vorlesung mit Übungen Visual Computing
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden
17. Prüfungsnummer/n und -name:	29501 Visual Computing (PL), schriftlich oder mündlich, Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündlichen 30 Min

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

Modul: 11330 Visualisierung

2. Modulkürzel:	051900011	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Daniel Weiskopf
9. Dozenten:	• Thomas Ertl • Daniel Weiskopf
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Wahlmodule aus Master Informatik BA (Komb) Informatik, 6. Semester → Module im Nebenfach → Katalog ISW
11. Voraussetzungen:	• 051900002 Computergraphik • 051900001 Mensch-Computer-Interaktion • 051240005 Numerik und Stochastik.
12. Lernziele:	Die Studierenden haben Wissen über Grundlagen, Algorithmen und Datenstrukturen für die Visualisierung sowie praktische Fähigkeiten durch die Arbeit mit Visualisierungssoftware erworben.
13. Inhalt:	Visualisierung behandelt alle Aspekte, die mit der visuellen Repräsentation von Daten aus wissenschaftlichen Experimenten, Simulationen, medizinischen Scannern, Datenbanken oder ähnlichen Datenquellen gewonnen werden, um zu einem tieferen Verständnis zu gelangen oder eine einfachere Darstellung komplexer Phänomene oder Sachverhalte zu erhalten. Um dieses Ziel zu erreichen, werden zum einen wohlbekannte Techniken aus dem Gebiet der interaktiven Computergraphik, zum anderen auch neu entwickelte Techniken angewendet. Entsprechend werden in dieser Vorlesung folgenden Themen behandelt: • Einführung, Historie, Visualisierungspipeline • Datenakquise und -repräsentation (Abtasten, Rekonstruktion, Gitter, Datenstrukturen) • Wahrnehmungsaspekte • Grundlegende Konzepte visueller Abbildungen • Visualisierung von Skalarfeldern (Isoflächenextraktion, Volumenrendering) • Visualisierung von Vektorfelder (Teilchenverfolgung, texturbasierte Methoden, Topologie) • Tensorfelder, Multiattributdaten • Hochdimensionale Daten und Informationsvisualisierung
14. Literatur:	• C. D. Hansen, C. R. Johnson, The Visualization Handbook, 2005 • C. Ware, Information Visualization: Perception for Design, 2004 • H. Schumann, W. Müller, Visualisierung: Grundlagen und allgemeine Methoden, 2000 • K. Engel, M. Hadwiger, J. M. Kniss, C. Rezk-Salama, D. Weiskopf, Real-time Volume Graphics, 2006
15. Lehrveranstaltungen und -formen:	• 113301 Vorlesung Visualisierung • 113302 Übungen Visualisierung

16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	11331 Visualisierung (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Maschinelle Sprachverarbeitung, 5. Semester → Wahlbereich E/I

400 Schlüsselqualifikationen fachaffin

Zugeordnete Module: 10290 Projekt-INF
10320 Seminar-INF

Modul: 10290 Projekt-INF

2. Modulkürzel:	051900095	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Thomas Ertl
9. Dozenten:	
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 5. Semester → Schlüsselqualifikationen fachaffin
11. Voraussetzungen:	• Basismodule der Informatik. Darüber hinaus variabel je nach Projektanforderung. (Wird per Aushang bekannt gegeben.)
12. Lernziele:	Die Teilnehmer können ein forschungsorientiertes Projekt unter Anleitung initiieren und planen. Sie können dazu notwendige Projektpläne erstellen, diese überwachen und ggf. den Realitäten anpassen. Sie können erforderliche Software beschaffen oder selbst erstellen. Sie verfügen insbesondere über die folgenden generischen Kompetenzen (Schlüsselqualifikationen): Sie können in Teams an einem gemeinsamen Vorhaben arbeiten und ihre Beiträge den übergeordneten Erfordernissen anpassen. Sie sind in der Lage, ihre Ergebnisse den Projektteilnehmern vorzustellen und zu diskutieren und sie dabei gegebenenfalls auch fachfremden Teilnehmern zu erläutern. Sie können moderne Präsentations- und Visualisierungstechniken erfolgreich einsetzen.
13. Inhalt:	Variabel: Es werden Seminare zu diversen häufig aktuellen Themen angeboten. Das Seminar INF kann in der Informatik oder in einem affinen Fach durchgeführt werden, wie etwa Computerlinguistik, Elektrotechnik, Mathematik oder Wirtschaftswissenschaften. Welche Seminare zugelassen sind, entscheidet die Studienkommission. Zugelassene Seminare werden typischer Weise durch Aushang bekannt gegeben. Die Seminare sind in Größe und Inhalt so gestaltet, dass die generischen Kompetenzen (Schlüsselqualifikationen) der Studierenden entwickelt werden.
14. Literatur:	• Literatur, die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben.
15. Lehrveranstaltungen und -formen:	102901 Seminar Projekt
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10291 Projekt-INF (USL), Sonstiges, Gewichtung: 0.0, Scheinkriterien: Aktive Teilnahme an den regelmäßigen Treffen und ein Projektbericht
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Technikpädagogik, 5. Semester

→ Hauptfach Informatik
→ Kernmodule Informatik

B.Sc. Technikpädagogik, 5. Semester
→ Wahlpflichtfach
→ Informatik

M.Sc. Technikpädagogik, 5. Semester
→ Wahlpflichtfach B
→ Affines Wahlpflichtfach Informatik
→ Affines Wahlpflichtfach Informatik Kernmodule

Modul: 10320 Seminar-INF

2. Modulkürzel:	050420095	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	2.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Volker Diekert
9. Dozenten:	• Dozenten der Informatik • Dozenten der Anorganischen Chemie
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 5. Semester → Schlüsselqualifikationen fachaffin
11. Voraussetzungen:	Basismodule der Informatik, darüber hinaus variabel: Je nach dem gewählten Seminarthema können Vorkenntnisse aus weiteren Vorlesungen benötigt werden.
12. Lernziele:	Die Studierenden können sich mit wissenschaftlicher Originalliteratur auseinandersetzen, deren Kernaussagen rezipieren und sich ein spezielles Thema überwiegend im Selbststudium erarbeiten. Sie sind fähig relevante Daten zu sammeln und zu interpretieren und ihre Erkenntnisse einem Fach- und Laienpublikum verständlich zu präsentieren und auf Fragen aus dem Publikum angemessen und sachgerecht zu reagieren. Sie haben gelernt, sich mit einem wissenschaftlichen Thema über einen längeren Zeitraum hinweg auseinander zu setzen und eigenständig aktuelle Hintergrundinformation zu beschaffen. Sie haben generische Kompetenzen erworben, etwa aktiv an einer wissenschaftlichen Diskussion zu einem vorher bekannten Thema teilzunehmen und durch Fragen an den Vortragenden ihr Verständnis zu erweitern. Sie können eine Diskussion leiten und moderieren und sind befähigt, ihre Ergebnisse den Seminarteilnehmern vorzustellen und mit Hilfe moderner Präsentationstechniken zu visualisieren.
13. Inhalt:	Variabel: Es werden Seminare zu diversen häufig aktuellen Themen angeboten. Das Seminar INF kann in der Informatik oder in einem affinen Fach durchgeführt werden, wie etwa Computerlinguistik, Elektrotechnik, Mathematik oder Wirtschaftswissenschaften. Welche Seminare zugelassen sind, entscheidet die Studienkommission. Zugelassene Seminare werden typischer Weise durch Aushang bekannt gegeben. Die Seminare sind in Größe und Inhalt so gestaltet, dass die generischen Kompetenzen (Schlüsselqualifikationen) der Studierenden entwickelt werden.
14. Literatur:	• Literatur, die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben.
15. Lehrveranstaltungen und -formen:	103201 Seminar
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 21 Stunden Nachbearbeitungszeit: 69 Stunden
17. Prüfungsnummer/n und -name:	10321 Seminar-INF (LBP), Sonstiges, Gewichtung: 1.0, Scheinkriterien sind in der Regel ein Vortrag, eine schriftliche Ausarbeitung, sowie die aktive Mitarbeit während der Seminarveranstaltung.
18. Grundlage für ... :	

19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Wirtschaftsinformatik, 6. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich II: Informatik Basis B.Sc. Technikpädagogik, 5. Semester → Hauptfach Informatik → Kernmodule Informatik Allgemein Lehramt (GymPO I) ab PO 2010 → Pflichtmodule

700 Zusatzmodule