

Modulhandbuch
Studiengang Bachelor of Science Informatik
Prüfungsordnung: 2009

Wintersemester 2016/17
Stand: 10. Oktober 2016

Universität Stuttgart
Keplerstr. 7
70174 Stuttgart

Kontaktpersonen:

Studiendekan/in:	Univ.-Prof. Stefan Wagner Institut für Softwaretechnologie Tel.: 0711/685-88455 E-Mail: stefan.wagner@informatik.uni-stuttgart.de
Studiengangsmanager/in:	Katrin Schneider Institut für Maschinelle Sprachverarbeitung Tel.: 685 88520 E-Mail: katrin.schneider@informatik.uni-stuttgart.de
Prüfungsausschussvorsitzende/r:	Apl. Prof. Ulrich Hertrampf Institut für Formale Methoden der Informatik Tel.: 7816-344 E-Mail: ulrich.hertrampf@f05.uni-stuttgart.de
Fachstudienberater/in:	Stefan Zimmer Institut für Parallele und Verteilte Systeme Tel.: E-Mail: stefan.zimmer@ipvs.uni-stuttgart.de
Stundenplanverantwortliche/r:	Apl. Prof. Ulrich Hertrampf Institut für Formale Methoden der Informatik Tel.: 7816-344 E-Mail: ulrich.hertrampf@f05.uni-stuttgart.de

Inhaltsverzeichnis

Präambel	6
Qualifikationsziele	7
100 Basismodule	8
12060 Datenstrukturen und Algorithmen	9
10190 Mathematik für Informatiker und Softwaretechniker	11
10260 Programmierkurs	13
10280 Programmierung und Software-Entwicklung	15
10930 Technische Grundlagen der Informatik	17
10940 Theoretische Grundlagen der Informatik	19
200 Kernmodule	21
10020 Algorithmik	22
14910 Berechenbarkeit und Komplexität	24
25610 Grundlagen des Software Engineerings	25
10210 Mensch-Computer-Interaktion	27
10220 Modellierung	29
10240 Numerische und Stochastische Grundlagen	31
10310 Rechnerorganisation	33
10330 Systemkonzepte und -programmierung	35
300 Ergänzungsmodule	37
320 Katalog ISG 1-3	38
10030 Architektur von Anwendungssystemen	39
10060 Computergraphik	41
10090 Grundlagen der Eingebetteten Systeme	43
10100 Grundlagen der Graphischen Ingenieursysteme	44
10110 Grundlagen der Künstlichen Intelligenz	46
56930 Grundlagen der Rechnerarchitektur	48
42410 Grundlagen des Wissenschaftlichen Rechnens	50
10170 Imaging Science	52
10180 Information Retrieval und Text Mining	54
39040 Rechnernetze	55
330 Katalog ISW 1-3	57
10030 Architektur von Anwendungssystemen	58
10060 Computergraphik	60
10100 Grundlagen der Graphischen Ingenieursysteme	62
10110 Grundlagen der Künstlichen Intelligenz	64
56930 Grundlagen der Rechnerarchitektur	66
42410 Grundlagen des Wissenschaftlichen Rechnens	68
10170 Imaging Science	70
10180 Information Retrieval und Text Mining	72
39040 Rechnernetze	73
340 Katalog ISW 4-7	75
31600 Machine learning for NLP	76
350 Wahlmodule aus Master Informatik	77
60860 3D Scanner - Algorithms and Systems	79
42910 Advanced Business Process Management	80
55600 Advanced Information Management	82
55740 Advanced Service Computing	83

29550 Algorithmische Geometrie	85
29760 Algorithmische Gruppentheorie	86
45760 Ausgewählte Kapitel der Algorithmentheorie	88
42480 Ausgewählte Kapitel des Wissenschaftlichen Rechnens	89
56680 Automaten über unendlichen Objekten	91
10040 Bildsynthese	93
42900 Business Process Management	95
57050 Compilerbau	97
29570 Computer Interface Technologien	99
29430 Computer Vision	100
29580 Data Compression	102
48480 Data Engineering	103
55620 Data Warehousing, Data Mining, and OLAP	105
29600 Digital System Design II	107
29590 Digitale Systeme	108
39250 Distributed Systems I	109
45730 Distributed Systems II	111
57680 Einführung in die Chaostheorie	113
29710 Embedded Systems Engineering	115
58190 Entwurf und Implementierung eines Compilers	117
56980 Fachpraktikum Cloud Architekturen und Management	118
29740 Fachpraktikum Eingebettete Systeme	119
29750 Fachpraktikum Rechnerarchitektur	121
24900 Fachpraktikum Rechnergestützte Methoden der Produktentwicklung	123
45750 Fachpraktikum Verteilte Systeme	125
58440 Fachpraktikum: Algorithmik	126
29440 Geometric Modeling and Computer Animation	127
29450 Graphentheorie	129
29610 Hardware Based Fault Tolerance	131
14380 Hardware Verification and Quality Assessment	133
42920 Hardware-Software-Codesign	135
42420 High Performance Computing	136
51720 IT-Strategy	138
48500 Image Synthesis	140
55610 Information Integration	142
55630 Information Visualization and Visual Analytics	144
60120 Interaktive Systeme	146
29460 Kryptographische Verfahren	147
29480 Loose Coupling and Message Based Applications	148
29470 Machine Learning	150
29640 Mikrocontroller	152
29720 Mobile Computing	155
10120 Modellbildung und Simulation	157
29730 Modelling, Simulation, and Specification	159
55650 Multimodal Interaction for Ubiquitous Computers	161
42460 Numerische Simulation	162
56790 Parallele Numerik	163
29650 Parallele Programmierung	165
10250 Parallele Systeme	166
48550 Practical Course Information Systems	167
48570 Practical Course Visual Computing	168
29660 Programmanalysen und Compilerbau	169
51740 Quantencomputing	171
29670 Rapid Prototyping	172
29680 Real-Time Programming	173
29690 Real-Time Video Processing I	175
29700 Real-Time Video Processing II	176
45740 Rechnernetze II	177

48620 Scientific Visualization	179
29510 Service Computing	181
46660 Service Management and Cloud Computing, and Evaluation	183
60140 Sprachbau mit Language Workbenches	185
29500 Visual Computing	187
400 Schlüsselqualifikationen fachaffin	189
10290 Projekt-INF	190
10320 Seminar-INF 1	192
42390 Seminar-INF 2	194
81100 Bachelorarbeit Informatik	196

Präambel

Informatik ist die Wissenschaft von der Informationsverarbeitung und den informationsverarbeitenden Systemen. Sie umfasst deren Theorie und Methodik, den Einsatz dieser Systeme, aber auch die Auswirkungen. Die Informatik ist damit ein Grundpfeiler der modernen Informationsgesellschaft. Informatiksysteme durchdringen unser tägliches Leben. Was noch vor wenigen Jahren unvorstellbar war, ist heute selbstverständlicher Standard. Die weltweite freie Bereitstellung von Wissen und die Möglichkeit, sich ohne Kosten per E-Mail auszutauschen sowie riesige Datenmengen, etwa in Form von Musik und Filmen zu speichern, bedeutet eine gesellschaftliche Neuerung, an deren Gestaltung man durch ein Informatikstudium aktiv mitwirken kann.

Durch Verfahren der Modellbildung und Abstraktion formuliert die Informatik allgemeine Gesetze, die der Informationsverarbeitung zugrunde liegen, und sucht Standardlösungen für praxisrelevante Aufgaben. Von wachsender Bedeutung wird dabei die Beherrschung immer komplexer werdender verteilter und vernetzter Systeme. Informatikerinnen und Informatiker operieren mit abstrakten Zeichen und Objekten, untersuchen Daten-, Sprach- und Systemstrukturen und entwickeln formale Programmiersprachen zur Formulierung von Algorithmen, Prozessen, Systemen und speziellen Anwendungen. Die Hard- und Software-Systeme stehen dabei als Forschungsobjekte und gleichzeitig als Werkzeuge im Mittelpunkt der Arbeit. Durch Visualisierung und Simulation werden neue Anwendungen erschlossen. Informatik ist einerseits eine Strukturwissenschaft, andererseits dominieren aber heute die ingenieurwissenschaftlichen Methoden und Verfahren. Die Informatik an der Universität Stuttgart ist geprägt durch hohen Praxisbezug und Anwendungen, ohne dabei die notwendigen Grundlagen zu vernachlässigen.

Auf den Bachelor-Studiengang Informatik bauen die Master-Studiengänge auf, die vom Fachbereich Informatik im Anschluss an das Bachelor-Studium angeboten werden. Dazu gehören neben den deutschsprachigen Masterstudiengängen *Informatik* und *Softwaretechnik* auch die englischsprachigen Masterstudiengänge *Computer Science* und *Computational Linguistics*. Es wird empfohlen, den Master als Abschluss eines universitären Studiums anzustreben.

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.

Qualifikationsziele

Das Curriculum des Studienganges sieht im 1. und 2. Semester eine Vertiefung der mathematischen und technischen Grundlagen der Informatik in Pflichtmodulen vor. Im 3. bis 6. Semester liegt der Schwerpunkt der Ausbildung auf diversen Grundlagen- und Spezialisierungsfächern verschiedener Bereiche der Informatik. Zusätzliche Inhalte sind Seminare, fachliche Module als Wahlpflichtbereich sowie fachübergreifende Schlüsselqualifikationen. Im 6. Semester liegt ein weiterer Schwerpunkt auf der Bachelorarbeit. Mit der Bachelorarbeit ist die Befähigung zu zeigen, innerhalb einer vorgegebenen Frist eine komplexe Aufgabenstellung aus dem Bereich der Informatik selbständig nach wissenschaftlichen Methoden zu bearbeiten und die Ergebnisse sachgerecht darzustellen.

Die Absolventinnen und Absolventen des Bachelorstudienganges Informatik

- verfügen über ein vertieftes mathematisches und ingenieurwissenschaftliches Wissen, welches sie befähigt, Probleme und Aufgabenstellungen der Informatik zu verstehen und kritisch einzuschätzen.
- sind in der Lage, jenes erlangte Wissen auf Erkenntnisse der Ingenieurwissenschaften anzuwenden.
- verfügen über Fachwissen auf den Gebieten der theoretischen, praktischen, technischen und angewandten Informatik und können Aufgabenstellungen der Informatik wissenschaftlich erkennen, beschreiben und bewerten, analysieren und lösen.
- haben umfassendes Verständnis über Softwareentwicklungsmethoden und Rechnersysteme, ihre Anwendungsmöglichkeiten und verfügen über die Fertigkeit, Konzepte und Lösungen für neue Softwaresysteme zu entwickeln.
- verfügen über eine verantwortliche und selbständige wissenschaftliche Arbeitsweise.

Die Beschäftigungsbereiche der Absolventinnen und Absolventen liegen u. a. in der Datenverarbeitungs- und Entwicklungsindustrie und in industriellen, kommerziellen, verwaltungstechnischen und wissenschaftlichen Anwendungen von Datenverarbeitungssystemen.

100 Basismodule

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

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:	Univ.-Prof. Daniel Weiskopf		
9. Dozenten:	• Andrés Bruhn • Thomas Ertl • Stefan Funke • Daniel Weiskopf		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 2. Semester → Basismodule B.Sc. Informatik, PO 2012, 2. Semester → Basismodule		
11. Empfohlene Voraussetzungen:	• Modul 10280 Programmierung und Software-Entwicklung		
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. Die Lernziele lassen sich wie folgt zusammenfassen: • 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		
13. Inhalt:	Es werden die folgenden Themen behandelt: • Vorgehensweise bei der Entwicklung und Implementierung von Algorithmen • Komplexität und Effizienz von Algorithmen, O-Notation • Listen (Stack, Queue, doppelt verkettete Listen) • Sortierverfahren (Selection-, Insertion-, Bubble-, Merge-, Quick-Sort) • Bäume (Binär-, AVL-, 2-3-4-, Rot-Schwarz-, B-Bäume, Suchbäume, Traversierung, Heap) • Räumliche Datenstrukturen (uniforme Gitter, Oktal-, BSP-, kD-, CSG-Bäume, Bounding-Volumes) • Graphen (Datenstrukturen, DFS, BFS, topologische Traversierung, Dijkstra-, A*-, Bellman-Ford-Algorithmen, minimale Spannbäume, maximaler Fluss) • Räumliche Graphen (Triangulierung, Voronoi, Delaunay, Graph-Layout) • Textalgorithmen (String-Matching, Knuth-Morris-Pratt, Boyer-Moore, reguläre Ausdrücke, Levenshtein-Distanz) • Hashing (Hashfunktionen, Kollisionen) • Verteilte Algorithmen (Petri-Netze, Programmieren nebenläufiger Abläufe, einige parallele und parallelisierte Algorithmen)		

	• Algorithmenentwurf und -muster (inkrementell, greedy, divide-and-conquer, dynamische Programmierung, Backtracking, randomisierte Algorithmen) • Maschinelles Lernen (überwachtes Lernen, Entscheidungsbäume, SVM, neuronale Netze; unüberwachtes Lernen, k-Means)								
14. Literatur:	• G. Saake, K. Sattler. <i>Algorithmen und Datenstrukturen: Eine Einführung mit Java</i> . 5. Auflage, dpunkt-Verlag, 2013 • T. Ottmann, P. Widmayer. <i>Algorithmen und Datenstrukturen</i> . 5. Auflage, Springer-Verlag, 2012								
15. Lehrveranstaltungen und -formen:	• 120601 Vorlesung Datenstrukturen und Algorithmen • 120602 Übung Datenstrukturen und Algorithmen								
16. Abschätzung Arbeitsaufwand:	<table> <tr> <td>Präsenzzeit:</td><td>63 h</td></tr> <tr> <td>Selbststudiums- /</td><td>207</td></tr> <tr> <td>Nachbearbeitungszeit:</td><td></td></tr> <tr> <td>Summe:</td><td>270 h</td></tr> </table>	Präsenzzeit:	63 h	Selbststudiums- /	207	Nachbearbeitungszeit:		Summe:	270 h
Präsenzzeit:	63 h								
Selbststudiums- /	207								
Nachbearbeitungszeit:									
Summe:	270 h								
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. • V Vorleistung (USL-V), schriftlich, eventuell mündlich								
18. Grundlage für ... :									
19. Medienform:									
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme								

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:	PD Andreas Markus Kollross		
9. Dozenten:	• Peter Lesky • Wolfgang Rump • Wolf-Patrick Düll • Andreas Markus Kollross		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 1. Semester → Basismodule B.Sc. Informatik, PO 2012, 1. Semester → Basismodule		
11. Empfohlene 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 Integralrechnung (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, Vorlesungsskripte, 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		

- V Übungsschein aus den beiden Veranstaltungen, jeweils im 1. oder 2. Fachsemester zu erwerben
Vorleistung (USL-V), schriftlich, eventuell mündlich
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

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:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Jonas Kuhn		
9. Dozenten:	Jason Utt		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 1. Semester → Basismodule		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Selbstständiges Erstellen von Programmen und Lösung von Programmieraufgaben in der Programmiersprache Python, mit einem Schwerpunkt auf Konzepten, die für die maschinelle Sprachverarbeitung und Computerlinguistik wichtig sind. -- Independently writing programs and solving programming tasks in the programming language Python, with emphasis on concepts relevant for Natural Language Processing and Computational Linguistics.		
13. Inhalt:	Das Modul "Programmierkurs" richtet sich primär an Studierende der Maschinellen Sprachverarbeitung (3. Semester), Computerlinguistik und Digital Humanities. Vermittelt werden die wichtigsten Konzepte der Programmiersprache Python und praktische Erfahrung bei der Erstellung von Python-Programmen bei der Verarbeitung von sprachlichen Daten und Ressourcen. Die Modulveranstaltung und die Materialien sind in der Regel überwiegend englischsprachig; es werden jedoch deutschsprachige Hilfestellungen angeboten. -- The module primarily targets students in Natural Language Processing (3rd semester), Computational Linguistics and Digital Humanities. It covers the key concepts of the programming language Python and provides practical experience in writing Python programs in the context of processing linguistic data and resources. Typically, the lectures of the module course as well as the materials are in English; however, students not fluent in English in the programming context will receive support in German.		
14. Literatur:	Folien. Slides.		
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. Criteria for credits are announced at the beginning of the course.		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von: Institut für Maschinelle Sprachverarbeitung

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:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 1. Semester → Basismodule B.Sc. Informatik, PO 2012, 1. Semester → Basismodule		
11. Empfohlene Voraussetzungen:	Keine. Teilnahme an einem Vorkurs Java ist hilfreich aber nicht notwendig.		
12. Lernziele:	Die Teilnehmer haben einen Überblick über das Gebiet der Informatik. Sie 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:	• Die Programmiersprache Java und die virtuelle Maschine • Objekte, Klassen, Schnittstellen, Blöcke, Programmstrukturen, Kontrakte • Klassenmodellierung mit der UML • Objekterzeugung und -ausführung • Boolesche Logik • Verzweigungen, Schleifen, Routinen, Abstraktionen, Modularisierung, Variablen, Zuweisungen • Rechner, Hardware • Syntaxdarstellungen • Übersicht über Programmiersprachen und -werkzeuge • Grundlegende Datenstrukturen und Algorithmen • Vererbung, Polymorphe • Semantik • Programmierung graphischer Oberflächen • Übergang zum Software Engineering		
14. Literatur:	• Appelrath, Hans-Jürgen und Ludewig, Jochen, "Skriptum Informatik - eine konventionelle Einführung", Verlag der Fachvereine Zürich und B.G. Teubner Stuttgart, 4. Auflage 1999 • Meyer, Bertrand, "Touch of Class", Springer-Verlag, 2009 • Savitch, Walter, "Java. An Introduction to Problem Solving and Programming", Pearson, 6. Auflage, 2012		
15. Lehrveranstaltungen und -formen:	• 102801 Vorlesung Programmierung und Softwareentwicklung • 102802 Übung Programmierung und Softwareentwicklung		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	63 h
	Selbststudiums- /	187 h
	Nachbearbeitungszeit:	
	Prüfungsvorbereitung:	20 h
	Summe:	270 h
17. Prüfungsnummer/n und -name:	• 10281 Programmierung und Software-Entwicklung (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, • V Vorleistung (USL-V), schriftlich, eventuell mündlich, Vorleistung: Übungsschein. Voraussetzungen werden zu Beginn vom Dozenten festgesetzt. Dazu gehören eine bestimmte Anzahl von Vorträgen in den Übungen und ein bestimmter Teil der Übungspunkte.	
18. Grundlage für ... :	12060 Datenstrukturen und Algorithmen	
19. Medienform:	• Folien über Beamer • Tafelanschrieb	
20. Angeboten von:	Software-Engineering	

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:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 1. Semester → Basismodule B.Sc. Informatik, PO 2012, 1. Semester → Basismodule		
11. Empfohlene Voraussetzungen:	Keine		
12. Lernziele:	• Elektrotechnische Grundlagen: Grundlegendes Verständnis elektrischer Schaltkreise und der Funktionsweise der Bauelemente und Komponenten von Computersystemen, wie Transistoren, digitale Halbleiterschaltungen, Speicher. • Digitaltechnische Komponenten: Fähigkeit zur Analyse, Konstruktion und Optimierung digitaler Schaltungen von begrenzter Komplexität.		
13. Inhalt:	1. Teil des Moduls (im Wintersemester, "Elektrotechnische Grundlagen"): • Informationsbegriff, Codierung, Darstellung mit analogen Größen • Übersicht über den Entwurf informationsverarbeitender Systeme • Boole'sche Algebra • Physikalische und mathematische Grundbegriffe der Elektrotechnik • Elektrostatisches Feld, Potential, Spannung und Kondensator • Elektrischer Strom, elektrische Netzwerke und Widerstand • Halbleitertechnik, Diode, Transistor • Digitale Grundsaltungen, Logik- und Speicherschaltungen 2. Teil des Moduls (im Sommersemester, "Digitaltechnische Komponenten"): • Schaltalgebra, Schaltnetze / kombinatorische Netzwerke • Verzögerungsanalyse • Kombinatorische Komponenten von Rechensystemen • Sequentielle Komponenten von Rechensystemen • Modelle sequentiellen Verhaltens, Schaltwerke / sequentielle Netzwerke • Taktung und Taktschemata • 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: 94 Stunden		

Nachbearbeitungszeit: 176 Stunden

17. Prüfungsnummer/n und -name:	• 10931 Technische Grundlagen der Informatik (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: In jeder Teilveranstaltung (1. Teil "Elektrotechnische Grundlagen" sowie 2. Teil "Digitaltechnische Komponenten") ist die aktive und erfolgreiche Teilnahme an einer Mindestzahl der Übungen erforderlich. Details werden zu Beginn der Lehrveranstaltung bekannt gegeben. • V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)

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:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	• Volker Diekert • Ulrich Hertrampf		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 1. Semester → Basismodule B.Sc. Informatik, PO 2012, 1. Semester → Basismodule		
11. Empfohlene 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: Semantik (Wahrheitswerte), Syntax (Axiome und Schlussregeln), Normalformen; Hornformeln; Endlichkeitssatz; aussagenlogische Resolution; • Einführung in die Prädikatenlogik 1. Stufe: Semantik und Syntax, Normalformen, Unifikatoren, Herbrand-Theorie, prädikatenlogische Resolution; • Elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, Euklidischer Algorithmus, Chinesischer Restsatz, Primzahltests, RSA-Verfahren; Wachstumsabschätzungen; Grundbegriffe der Wahrscheinlichkeitsrechnung; Kombinatorik; Graphen. 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 h
	Selbststudiums- /	276 h
	Nachbearbeitungszeit:	
	Summe:	360 h
17. Prüfungsnummer/n und -name:	• 10941 Theoretische Grundlagen der Informatik (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min.	
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:	Institut für Formale Methoden der Informatik	

200 Kernmodule

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

Modul: 10020 Algorithmik

2. Modulkürzel:	050420015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	• Volker Diekert • Stefan Funke • Ulrich Hertrampf		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Kernmodule B.Sc. Informatik, PO 2012, 4. Semester → Kernmodule		
11. Empfohlene 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 D. Ullman: The Design and Analysis of Computer Algorithms, 1974 • Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullmann: Data Structures and 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 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	• 10021 Algorithmik (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvoraussetzung: Übungsschein • V Vorleistung (USL-V), schriftlich, eventuell mündlich		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von: Institut für Formale Methoden der Informatik

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:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	• Stefan Funke • Volker Diekert • Ulrich Hertrampf		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 3. Semester → Kernmodule B.Sc. Informatik, PO 2012, 3. Semester → Kernmodule		
11. Empfohlene 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 verschiedenen Konkretisierungen des Algorithmienbegriffs, Churchs These, Grenzen zwischen Entscheidbarkeit 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: 42 h Selbststudiums- / 118 h Nachbearbeitungszeit: Prüfungsvorbereitung: 20 h Summe: 180 h		
17. Prüfungsnummer/n und -name:	• 14911 Berechenbarkeit und Komplexität (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein • V Vorleistung (USL-V), schriftlich, eventuell mündlich		
18. Grundlage für ... :	10020 Algorithmik		
19. Medienform:			
20. Angeboten von:	Institut für Formale Methoden der Informatik		

Modul: 25610 Grundlagen des Software Engineerings

2. Modulkürzel:	51520170	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Stefan Wagner		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Kernmodule B.Sc. Informatik, PO 2012, 5. Semester → Kernmodule		
11. Empfohlene Voraussetzungen:	• Modul 10280 Programmierung und Software-Entwicklung • Modul 12060 Datenstrukturen und Algorithmen • Programmiererfahrung		
12. Lernziele:	Die Teilnehmer kennen die Grundbegriffe des Software Engineerings und haben einen Überblick über die Methoden und Techniken, die dort angewandt werden. Einige ausgewählte Methoden und Techniken können angewandt werden.		
13. Inhalt:	Software Engineering kann in einer Vorlesung nicht erschöpfend behandelt werden. GSE gibt einen Überblick über das Gebiet und vertieft einzelne Themen, damit diese in der Praxis verwendet werden können. Es bildet damit auch die Basis für weitere Vertiefungen in diesem Gebiet. Die Vorlesung behandelt technische und andere Aspekte der Softwarebearbeitung. Die einzelnen Themen sind: • Geschichte und Konzepte des Software Engineerings • Der Software-Lebenszyklus und Software-Management • Software-Prüfung und Qualitätssicherung • Methoden, Sprachen und Werkzeuge für die einzelnen Phasen: Spezifikation, Grobentwurf, Feinentwurf, Implementierung, Test Viele dieser Aspekte werden speziell mit Bezug auf agile Softwareentwicklung am Beispiel Scrum diskutiert. Dieses Modul kommt, wenn die Voraussetzungen erfüllt sind, auch für andere Fachrichtungen in Frage.		
14. Literatur:	• Ludewig, Lichter: Software Engineering. 2. Aufl. dpunkt-Verlag, 2010 • Pfleeger, Atlee: Software Engineering. Pearson, 2010 • Rubin: Essential Scrum. Addison-Wesley, 2013		
15. Lehrveranstaltungen und -formen:	• 256101 Vorlesung Grundlagen des Software Engineerings • 256102 Übung Grundlagen des Software Engineerings		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	25611 Grundlagen des Software Engineerings (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:	• Folien am Beamer unterstützt durch Tafel und Overhead		

- Dokumente, Links und Diskussionsforen in ILIAS

20. Angeboten von:

Software-Engineering

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:	Univ.-Prof. Albrecht Schmidt		
9. Dozenten:	• Albrecht Schmidt • Thomas Ertl • Daniel Weiskopf • Niels Henze		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Kernmodule B.Sc. Informatik, PO 2012, 4. Semester → Kernmodule		
11. Empfohlene Voraussetzungen:	• Modul 10280 Programmierung und Software-Entwicklung		
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 Dachsel. 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:	<table> <tr> <td>Präsenzzeit:</td><td>42 h</td></tr> <tr> <td>Selbststudiums- /</td><td>138 h</td></tr> <tr> <td>Nachbearbeitungszeit:</td><td></td></tr> <tr> <td>Summe:</td><td>180 h</td></tr> </table>	Präsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Präsenzzeit:	42 h								
Selbststudiums- /	138 h								
Nachbearbeitungszeit:									
Summe:	180 h								
17. Prüfungsnummer/n und -name:	• 102111 Mensch-Computer-Interaktion (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein • V Vorleistung (USL-V), schriftlich, eventuell mündlich								
18. Grundlage für ... :									
19. Medienform:									
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme								

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:	Univ.-Prof. Frank Leymann		
9. Dozenten:	• Bernhard Mitschang • Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Kernmodule B.Sc. Informatik, PO 2012, 4. Semester → Kernmodule		
11. Empfohlene 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), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein • V Vorleistung (USL-V), schriftlich, eventuell mündlich		
18. Grundlage für ... :	• 10030 Architektur von Anwendungssystemen		

- 10080 Datenbanken und Informationssysteme
-

19. Medienform:

20. Angeboten von: Institut für Architektur von Anwendungssystemen

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:	Dirk Pflüger		
9. Dozenten:	• Dirk Pflüger • Stefan Zimmer • Miriam Mehl		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 3. Semester → Kernmodule B.Sc. Informatik, PO 2012, 3. Semester → Kernmodule		
11. Empfohlene Voraussetzungen:	• Modul 10190 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 stochastischen 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 Gleitpunktarithmetik • 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:	• Dahmen, Reusken; Numerik für Ingenieure • Schwarz, Köckler; Numerische Mathematik • Huckle, Schneider; Numerik für Informatiker • Henze; Stochastik für Einsteiger • Schickinger, Steger; Diskrete Strukturen, Band 2		
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 h
	Selbststudiums- /	207 h
	Nachbearbeitungszeit:	
	Summe:	270 h
17. Prüfungsnummer/n und -name:	• 10241 Numerische und Stochastische Grundlagen (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein • V Vorleistung (USL-V), schriftlich, eventuell mündlich	
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:	Simulation großer Systeme	

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:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 3. Semester → Kernmodule		
11. Empfohlene Voraussetzungen:	- Für Rechnerorganisation 1: Technische Grundlagen der Informatik (10930) - 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, - Kenntnisse 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: Cachelstrukturen 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 Grundsaltungen 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:	• 103111 Rechnerorganisation 1 (PL), schriftliche Prüfung, 90 Min., Gewichtung: 7.0 • 103112 Rechnerorganisation 1 - Praktikum und Übungsaufgaben (LBP), Studienbegleitend, Gewichtung: 3.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	• 10090 Grundlagen der Eingebetteten Systeme • 10140 Advanced Processor Architecture • 11900 Design and Test of Systems-on-a-Chip • 14380 Hardware Verification and Quality Assessment • 29560 Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems • 29610 Hardware Based Fault Tolerance
19. Medienform:	
20. Angeboten von:	

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:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	• Kurt Rothermel • Frank Dürr		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 3. Semester → Kernmodule		
11. Empfohlene 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 Basialgorithmen 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), schriftliche Prüfung, 90 Min., Gewichtung: 7.0 • 10332 Systemkonzepte und -programmierung - Übungsschein (LBP), schriftliche Prüfung, 30 Min., Gewichtung: 3.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

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:	10030	Architektur von Anwendungssystemen
	10060	Computergraphik
	10090	Grundlagen der Eingebetteten Systeme
	10100	Grundlagen der Graphischen Ingenieursysteme
	10110	Grundlagen der Künstlichen Intelligenz
	10170	Imaging Science
	10180	Information Retrieval und Text Mining
	39040	Rechnernetze
	42410	Grundlagen des Wissenschaftlichen Rechnens
	56930	Grundlagen der Rechnerarchitektur

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:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	Vorlesungen des Grundstudiums.		
12. Lernziele:	Die Vorlesung erläutert den Begriff der Architektur von Anwendungssystemen und die Rolle des Architekten solcher Systeme. Die wesentlichen Bestandteile von Anwendungsarchitektur wie etwa Datenbanksysteme, Anwendungsserver, Messaging Systeme, Workflowsysteme und TP-Monitore werden diskutiert. Die wesentlichen Mustern zur Erstellung von Anwendungssystemen sind verstanden.		
13. Inhalt:	Architekturelle Stile wie etwa N-stufige Aufbauten oder Service-Orientierung werden vorgestellt. Architekturmuster werden detailliert. Fundamentale Konzepte wie Transaktionen und Queuing werden eingeführt. Darauf aufbauend wird Direct TP vs Queues TP diskutiert. Grundlegende Qualitätseigenschaften wie Verfügbarkeit und Skalierbarkeit werden erläutert und Mechanismen zu deren Erzielen eingeführt. Die Rolle von Komponenten und Programmierung im Großen wird heraus gearbeitet und Modell-getriebene Architektur vorgestellt.		
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, Patterns 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,
18. Grundlage für ... :	• 29530 Business Process Management • 29480 Loose Coupling and Message Based Applications • 29510 Service Computing
19. Medienform:	Vorlesungen mit begleitenden Übungen
20. Angeboten von:	Architektur von Anwendungssystemen

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:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	• Thomas Ertl • Daniel Weiskopf • Guido Reina		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	• Modul 10210 Mensch-Computer-Interaktion • Modul 41590 Einführung in die 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 Prozess der Bildsynthese • Graphische Geräte, visuelle Wahrnehmung, Farbsysteme • Grundlegende Rastergraphik und Bildverarbeitung • Raytracing und Beleuchtungsmodelle • 2D und 3D Geometrietransformationen, 3D Projektion • Graphikprogrammierung in OpenGL 3 • Texturen • Polygonale und hierarchische Modelle • Rasterisierung und Verdeckungsberechnung • Grundlagen der geometrischen Modellierung (Kurven, Flächen) • Räumliche Datenstrukturen 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 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
17. Prüfungsnummer/n und -name:	• 10061 Computergraphik (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein. • V Vorleistung (USL-V), schriftlich, eventuell mündlich	
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme	

Modul: 10090 Grundlagen der Eingebetteten Systeme

2. Modulkürzel:	051711010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 →		
11. Empfohlene Voraussetzungen:	• Modul 051700005 Rechnerorganisation		
12. Lernziele:	Methodische und werkzeugunterstützte Konstruktion von Hardware und Software für auf spezifische Anwendungen hin optimierte eingebettete Systeme.		
13. Inhalt:	Einführung in Charakteristik und Anwendungsgebiete eingebetteter Systeme, hardwarenahe Softwareentwicklung, Software-Scheduling, eingebettete Algorithmen (digitale Signalverarbeitung, Kanal- und Quellencodierung am Beispiel Viterbi-Algorithmus und MPEG-Video-Codec), zustandsbasierte Modellierung (Statecharts), eingebettete Prozessoren (Microcontroller, digitale Signalprozessoren, ARM), Bussysteme und Speicher, Systemsynthese (Taskgraphen, Allokation, Bindung, Ablaufplanung/Scheduling)		
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), schriftliche Prüfung, 90 Min., Gewichtung: 75.0 • 10092 Grundlagen der Eingebetteten Systeme - Rechnerübungen (LBP), Sonstiges, Gewichtung: 25.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)		

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:	Univ.-Prof. Dieter Roller		
9. Dozenten:	Dieter Roller		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene 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 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
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:

Institut für Rechnergestützte Ingenieursysteme

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:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	• Andrés Bruhn • Marc Toussaint		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene 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		
14. Literatur:	• S. Russell, P. Norvig, Künstliche Intelligenz, 2004 • G. F. Luger, Künstliche Intelligenz, 2001		
15. Lehrveranstaltungen und -formen:	• 101101 Vorlesung Grundlagen der Künstlichen Intelligenz • 101102 Übung Grundlagen der Künstlichen Intelligenz		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		

17. Prüfungsnummer/n und -name:
- 10111 Grundlagen der Künstlichen Intelligenz (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben
 - V Vorleistung (USL-V), schriftlich, eventuell mündlich
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

Modul: 56930 Grundlagen der Rechnerarchitektur

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	• Michael Kochte • Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	• Modul 41930 Rechnerorganisation		
12. Lernziele:	• Tiefes Verständnis von Entwurfskonzepten, die in modernen Prozessoren und Rechensystemen Verwendung finden • Kenntnis von Entwurfsherausforderungen. • Verständnis von aktuellen und zukünftigen Entwicklungstrends		
13. Inhalt:	Grundlegende und fortgeschrittene Themen der Rechnerarchitektur, inklusive: • Technologiegrundlagen: Entwurfsverfahren, Herstellungsmethoden, Ausbeute, Test und Zuverlässigkeit, Kosten und Qualität, Skalierung. • Rechen- und Verlustleistung: Analyse und Optimierung • Rechnerarithmetik: Effiziente Hardwarestrukturen für grundlegende Arithmetik, Implementierung von Logarithmen, Exponentialfunktion und trigonometrischen Funktionen, arithmetische Pipelines, praktische Implementierungen von Gleitkommaarithmetik (Cell SPE, SPARC). • Instruktionsparallelismus (ILP): Superskalarität, statisches und dynamisches Scheduling, out-of-order execution, VLIW Prozessoren, Multithreading. • Datenparallelismus (DLP): Vektorprozessoren, SIMD, Grafikprozessoreinheiten (GPGPU) • Threadparallelismus (TLP): Mehrprozessorsysteme, Speicherkohärenz und Synchronisierung. • Speicher- und Cache-Architekturen: Entwurf und Optimierung		
14. Literatur:	• J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 2012 • I. Koren, Computer Arithmetic Algorithms, 2001 • Powerpoint Foliensatz • Auswahl von wissenschaftlichen Artikeln		
15. Lehrveranstaltungen und -formen:	• 569301 Vorlesung Grundlagen der Rechnerarchitektur • 569302 Übung Grundlagen der Rechnerarchitektur		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
17. Prüfungsnummer/n und -name:	56931 Grundlagen der Rechnerarchitektur (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0	
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:	Rechnerarchitektur	

Modul: 42410 Grundlagen des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	• Dirk Pflüger • Stefan Zimmer • Miriam Mehl		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	• Modul 10190 Mathematik für Informatiker und Softwaretechniker und • Modul 10240 Numerische und Stochastische Grundlagen der Informatik bzw. • Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker		
12. Lernziele:	Kenntnis grundlegender Konzepte, Algorithmen und Methoden des Wissenschaftlichen Rechnens. Fähigkeit, mit den erlernten Kenntnissen selbständig Methoden zu entwickeln, zu analysieren und umzusetzen, mit denen anwendungsorientierte Probleme effizient und genau gelöst werden können.		
13. Inhalt:	• Überblick über die Simulationspipeline und die Wechselwirkungen zwischen den einzelnen Schritten • Skalenabhängige Modellierung • Diskretisierung (Gitter, Finite Elemente, Zeitschrittverfahren) • Algorithmen (Gittergenerierung, Adaptivität, Lineare Löser, Linked-Cell, Fast Multipole) • Parallelisierung (Gitterpartitionierung, Lastbalancierung) • Kurzer Überblick über die Visualisierung		
14. Literatur:	Martin Hanke-Bourgeois: Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens; Vieweg+Teubner Verlag 2009		
15. Lehrveranstaltungen und -formen:	• 424101 Vorlesung Grundlagen des Wissenschaftlichen Rechnens • 424102 Übung Grundlagen des Wissenschaftlichen Rechnens		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	42411 Grundlagen des Wissenschaftlichen Rechnens (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Simulation großer Systeme

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:	Univ.-Prof. Andrés Bruhn		
9. Dozenten:	Andrés Bruhn		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	• Modul 10190 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 • 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. MPEG) • Bildverbesserung und Restauration • Elementare Segmentierungsverfahren • Fundamentals of optics such as pinhole camera and lens equation • Image acquisition: Cameras, lenses, illumination, acquisition process • Image representation: Discretization, color spaces • 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. mpeg) • Image enhancement and restauration • Basics of segmentation								
14. Literatur:	• Bässmann, Henning; Kreys, Jutta: Bildverarbeitung Ad Oculos, 2004 • Forsyth, David and Ponce, Jean: Computer Vision. A Modern Approach, 2003 • Gonzalez, Rafael C.; Woods, Richard E.; Eddins, Steven L.: Digital Image Processing, 2004 • Bigun, J.: Vision with Direction, 2006 • 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:	<table> <tr> <td>Präsenzzeit:</td><td>42 h</td></tr> <tr> <td>Selbststudiums- /</td><td>138 h</td></tr> <tr> <td>Nachbearbeitungszeit:</td><td></td></tr> <tr> <td>Summe:</td><td>180 h</td></tr> </table>	Präsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Präsenzzeit:	42 h								
Selbststudiums- /	138 h								
Nachbearbeitungszeit:									
Summe:	180 h								
17. Prüfungsnummer/n und -name:	• 10171 Imaging Science (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben. • V Vorleistung (USL-V), schriftlich, eventuell mündlich								
18. Grundlage für ... :	• 29430 Computer Vision • 55640 Correspondence Problems in Computer Vision								
19. Medienform:									
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme								

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:	Roman Klinger		
9. Dozenten:	Roman Klinger		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	Erfahrung mit Programmierung und Unix, erster Kontakt mit Verfahren des Maschinellen Lernens		
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 • 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 • 10182 Information Retrieval und Text Mining - Hausübungen (USL), Sonstiges, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Maschinelle Sprachverarbeitung		

Modul: 39040 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:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	• Kurt Rothermel • Frank Dürr		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	• 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen • Grundkenntnisse in Java		
12. Lernziele:	• Versteht grundsätzliche Eigenschaften, Konzepte und Methoden von Rechnernetzen, insbesondere 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 netzgestützten 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:

- 390401 VL Rechnernetze
- 390402 ÜB Rechnernetze

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden
Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name:

- 39041 Rechnernetze (PL), schriftlich oder mündlich, 90 Min.,
Gewichtung: 1.0, Prüfungsdauer: 90 min schriftlich oder 30
min mündlich Exam duration: 90 min written exam or 30 min
oral exam
- V Vorleistung (USL-V), schriftlich, eventuell mündlich

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Verteilte Systeme

330 Katalog ISW 1-3

Zugeordnete Module:	10030	Architektur von Anwendungssystemen
	10060	Computergraphik
	10100	Grundlagen der Graphischen Ingenieursysteme
	10110	Grundlagen der Künstlichen Intelligenz
	10170	Imaging Science
	10180	Information Retrieval und Text Mining
	39040	Rechnernetze
	42410	Grundlagen des Wissenschaftlichen Rechnens
	56930	Grundlagen der Rechnerarchitektur

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:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	Vorlesungen des Grundstudiums.		
12. Lernziele:	Die Vorlesung erläutert den Begriff der Architektur von Anwendungssystemen und die Rolle des Architekten solcher Systeme. Die wesentlichen Bestandteile von Anwendungsarchitektur wie etwa Datenbanksysteme, Anwendungsserver, Messaging Systeme, Workflowsysteme und TP-Monitore werden diskutiert. Die wesentlichen Mustern zur Erstellung von Anwendungssystemen sind verstanden.		
13. Inhalt:	Architekturelle Stile wie etwa N-stufige Aufbauten oder Service-Orientierung werden vorgestellt. Architekturmuster werden detailliert. Fundamentale Konzepte wie Transaktionen und Queuing werden eingeführt. Darauf aufbauend wird Direct TP vs Queues TP diskutiert. Grundlegende Qualitätseigenschaften wie Verfügbarkeit und Skalierbarkeit werden erläutert und Mechanismen zu deren Erzielen eingeführt. Die Rolle von Komponenten und Programmierung im Großen wird heraus gearbeitet und Modell-getriebene Architektur vorgestellt.		
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,
18. Grundlage für ... :	• 29530 Business Process Management • 29480 Loose Coupling and Message Based Applications • 29510 Service Computing
19. Medienform:	Vorlesungen mit begleitenden Übungen
20. Angeboten von:	Architektur von Anwendungssystemen

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:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	• Thomas Ertl • Daniel Weiskopf • Guido Reina		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	• Modul 10210 Mensch-Computer-Interaktion • Modul 41590 Einführung in die 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 Prozess der Bildsynthese • Graphische Geräte, visuelle Wahrnehmung, Farbsysteme • Grundlegende Rastergraphik und Bildverarbeitung • Raytracing und Beleuchtungsmodelle • 2D und 3D Geometrietransformationen, 3D Projektion • Graphikprogrammierung in OpenGL 3 • Texturen • Polygonale und hierarchische Modelle • Rasterisierung und Verdeckungsberechnung • Grundlagen der geometrischen Modellierung (Kurven, Flächen) • Räumliche Datenstrukturen 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 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
17. Prüfungsnummer/n und -name:	• 10061 Computergraphik (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein. • V Vorleistung (USL-V), schriftlich, eventuell mündlich	
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme	

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:	Univ.-Prof. Dieter Roller		
9. Dozenten:	Dieter Roller		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene 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 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
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: Institut für Rechnergestützte Ingenieursysteme

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:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	• Andrés Bruhn • Marc Toussaint		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene 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		
14. Literatur:	• S. Russell, P. Norvig, Künstliche Intelligenz, 2004 • G. F. Luger, Künstliche Intelligenz, 2001		
15. Lehrveranstaltungen und -formen:	• 101101 Vorlesung Grundlagen der Künstlichen Intelligenz • 101102 Übung Grundlagen der Künstlichen Intelligenz		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		

17. Prüfungsnummer/n und -name:
- 10111 Grundlagen der Künstlichen Intelligenz (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben
 - V Vorleistung (USL-V), schriftlich, eventuell mündlich
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

Modul: 56930 Grundlagen der Rechnerarchitektur

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	• Michael Kochte • Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	• Modul 41930 Rechnerorganisation		
12. Lernziele:	• Tiefes Verständnis von Entwurfskonzepten, die in modernen Prozessoren und Rechensystemen Verwendung finden • Kenntnis von Entwurfsherausforderungen. • Verständnis von aktuellen und zukünftigen Entwicklungstrends		
13. Inhalt:	Grundlegende und fortgeschrittene Themen der Rechnerarchitektur, inklusive: • Technologiegrundlagen: Entwurfsverfahren, Herstellungsmethoden, Ausbeute, Test und Zuverlässigkeit, Kosten und Qualität, Skalierung. • Rechen- und Verlustleistung: Analyse und Optimierung • Rechnerarithmetik: Effiziente Hardwarestrukturen für grundlegende Arithmetik, Implementierung von Logarithmen, Exponentialfunktion und trigonometrischen Funktionen, arithmetische Pipelines, praktische Implementierungen von Gleitkommaarithmetik (Cell SPE, SPARC). • Instruktionsparallelismus (ILP): Superskalarität, statisches und dynamisches Scheduling, out-of-order execution, VLIW Prozessoren, Multithreading. • Datenparallelismus (DLP): Vektorprozessoren, SIMD, Grafikprozessoreinheiten (GPGPU) • Threadparallelismus (TLP): Mehrprozessorsysteme, Speicherkohärenz und Synchronisierung. • Speicher- und Cache-Architekturen: Entwurf und Optimierung		
14. Literatur:	• J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 2012 • I. Koren, Computer Arithmetic Algorithms, 2001 • Powerpoint Foliensatz • Auswahl von wissenschaftlichen Artikeln		
15. Lehrveranstaltungen und -formen:	• 569301 Vorlesung Grundlagen der Rechnerarchitektur • 569302 Übung Grundlagen der Rechnerarchitektur		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
17. Prüfungsnummer/n und -name:	56931 Grundlagen der Rechnerarchitektur (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0	
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:	Rechnerarchitektur	

Modul: 42410 Grundlagen des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	• Dirk Pflüger • Stefan Zimmer • Miriam Mehl		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	• Modul 10190 Mathematik für Informatiker und Softwaretechniker und • Modul 10240 Numerische und Stochastische Grundlagen der Informatik bzw. • Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker		
12. Lernziele:	Kenntnis grundlegender Konzepte, Algorithmen und Methoden des Wissenschaftlichen Rechnens. Fähigkeit, mit den erlernten Kenntnissen selbständig Methoden zu entwickeln, zu analysieren und umzusetzen, mit denen anwendungsorientierte Probleme effizient und genau gelöst werden können.		
13. Inhalt:	• Überblick über die Simulationspipeline und die Wechselwirkungen zwischen den einzelnen Schritten • Skalenabhängige Modellierung • Diskretisierung (Gitter, Finite Elemente, Zeitschrittverfahren) • Algorithmen (Gittergenerierung, Adaptivität, Lineare Löser, Linked-Cell, Fast Multipole) • Parallelisierung (Gitterpartitionierung, Lastbalancierung) • Kurzer Überblick über die Visualisierung		
14. Literatur:	Martin Hanke-Bourgeois: Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens; Vieweg+Teubner Verlag 2009		
15. Lehrveranstaltungen und -formen:	• 424101 Vorlesung Grundlagen des Wissenschaftlichen Rechnens • 424102 Übung Grundlagen des Wissenschaftlichen Rechnens		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	42411 Grundlagen des Wissenschaftlichen Rechnens (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Simulation großer Systeme

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:	Univ.-Prof. Andrés Bruhn		
9. Dozenten:	Andrés Bruhn		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	• Modul 10190 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 • 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. MPEG) • Bildverbesserung und Restauration • Elementare Segmentierungsverfahren • Fundamentals of optics such as pinhole camera and lens equation • Image acquisition: Cameras, lenses, illumination, acquisition process • Image representation: Discretization, color spaces • 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. mpeg) • Image enhancement and restauration • Basics of segmentation								
14. Literatur:	• Bässmann, Henning; Kreys, Jutta: Bildverarbeitung Ad Oculos, 2004 • Forsyth, David and Ponce, Jean: Computer Vision. A Modern Approach, 2003 • Gonzalez, Rafael C.; Woods, Richard E.; Eddins, Steven L.: Digital Image Processing, 2004 • Bigun, J.: Vision with Direction, 2006 • 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:	<table> <tr> <td>Präsenzzeit:</td><td>42 h</td></tr> <tr> <td>Selbststudiums- /</td><td>138 h</td></tr> <tr> <td>Nachbearbeitungszeit:</td><td></td></tr> <tr> <td>Summe:</td><td>180 h</td></tr> </table>	Präsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Präsenzzeit:	42 h								
Selbststudiums- /	138 h								
Nachbearbeitungszeit:									
Summe:	180 h								
17. Prüfungsnummer/n und -name:	• 10171 Imaging Science (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben. • V Vorleistung (USL-V), schriftlich, eventuell mündlich								
18. Grundlage für ... :	• 29430 Computer Vision • 55640 Correspondence Problems in Computer Vision								
19. Medienform:									
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme								

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:	Roman Klinger		
9. Dozenten:	Roman Klinger		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	Erfahrung mit Programmierung und Unix, erster Kontakt mit Verfahren des Maschinellen Lernens		
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 • 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 • 10182 Information Retrieval und Text Mining - Hausübungen (USL), Sonstiges, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Maschinelle Sprachverarbeitung		

Modul: 39040 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:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	• Kurt Rothermel • Frank Dürr		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISG 1-3 → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Katalog ISW 1-3 →		
11. Empfohlene Voraussetzungen:	• 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen • Grundkenntnisse in Java		
12. Lernziele:	• Versteht grundsätzliche Eigenschaften, Konzepte und Methoden von Rechnernetzen, insbesondere 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 netzgestützten 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:

- 390401 VL Rechnernetze
 - 390402 ÜB Rechnernetze
-

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden
Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name:

- 39041 Rechnernetze (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Prüfungsdauer: 90 min schriftlich oder 30 min mündlich Exam duration: 90 min written exam or 30 min oral exam
 - V Vorleistung (USL-V), schriftlich, eventuell mündlich
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Verteilte Systeme

340 Katalog ISW 4-7

Zugeordnete Module: 31600 Machine learning for NLP

Modul: 31600 Machine learning for NLP

2. Modulkürzel:	052400616	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	2.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Sebastian Pado		
9. Dozenten:	Sebastian Pado		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Katalog ISW 4-7 → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Katalog ISW 4 →		
11. Empfohlene Voraussetzungen:	Statistical natural language processing (recommended)		
12. Lernziele:	Students have acquired in-depth knowledge of several machine learning methods that are used in natural language processing and are familiar with the relevant literature.		
13. Inhalt:	- Maximum entropy models - Regression and regularized regression - Support vector machines - Sequence models - Generative models - Parameter estimation		
14. Literatur:	Abney, Semisupervised Learning for Computational Linguistics, Chapman and Hall/CRC, 2007. Bishop, Pattern Recognition and Machine Learning, Springer, 2007.		
15. Lehrveranstaltungen und -formen:	316001 Seminar course Machine learning for NLP		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 28h Selbststudium: 60h		
17. Prüfungsnummer/n und -name:	31601 Machine learning for NLP (BSL), schriftlich, eventuell mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Maschinelle Sprachverarbeitung		

350 Wahlmodule aus Master Informatik

Zugeordnete Module:	10040	Bildsynthese
	10120	Modellbildung und Simulation
	10250	Parallele Systeme
	14380	Hardware Verification and Quality Assessment
	24900	Fachpraktikum Rechnergestützte Methoden der Produktentwicklung
	29430	Computer Vision
	29440	Geometric Modeling and Computer Animation
	29450	Graphentheorie
	29460	Kryptographische Verfahren
	29470	Machine Learning
	29480	Loose Coupling and Message Based Applications
	29500	Visual Computing
	29510	Service Computing
	29550	Algorithmische Geometrie
	29570	Computer Interface Technologien
	29580	Data Compression
	29590	Digitale Systeme
	29600	Digital System Design II
	29610	Hardware Based Fault Tolerance
	29640	Mikrocontroller
	29650	Parallele Programmierung
	29660	Programmanalysen und Compilerbau
	29670	Rapid Prototyping
	29680	Real-Time Programming
	29690	Real-Time Video Processing I
	29700	Real-Time Video Processing II
	29710	Embedded Systems Engineering
	29720	Mobile Computing
	29730	Modelling, Simulation, and Specification
	29740	Fachpraktikum Eingebettete Systeme
	29750	Fachpraktikum Rechnerarchitektur
	29760	Algorithmische Gruppentheorie
	39250	Distributed Systems I
	42420	High Performance Computing
	42460	Numerische Simulation
	42480	Ausgewählte Kapitel des Wissenschaftlichen Rechnens
	42900	Business Process Management
	42910	Advanced Business Process Management
	42920	Hardware-Software-Codesign
	45730	Distributed Systems II
	45740	Rechnernetze II
	45750	Fachpraktikum Verteilte Systeme
	45760	Ausgewählte Kapitel der Algorithmentheorie
	46660	Service Management and Cloud Computing, and Evaluation
	48480	Data Engineering
	48500	Image Synthesis
	48550	Practical Course Information Systems
	48570	Practical Course Visual Computing
	48620	Scientific Visualization
	51720	IT-Strategy
	51740	Quantencomputing
	55600	Advanced Information Management
	55610	Information Integration
	55620	Data Warehousing, Data Mining, and OLAP

55630 Information Visualization and Visual Analytics
55650 Multimodal Interaction for Ubiquitous Computers
55740 Advanced Service Computing
56680 Automaten über unendlichen Objekten
56790 Parallele Numerik
56980 Fachpraktikum Cloud Architekturen und Management
57050 Compilerbau
57680 Einführung in die Chaostheorie
58190 Entwurf und Implementierung eines Compilers
58440 Fachpraktikum: Algorithmik
60120 Interaktive Systeme
60140 Sprachbau mit Language Workbenches
60860 3D Scanner - Algorithms and Systems

Modul: 60860 3D Scanner - Algorithms and Systems

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Univ.-Prof. Sven Simon	
9. Dozenten:			
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:			
12. Lernziele:			
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:	608601	Vorlesung mit Übung 3D-Scanner - Algorithmen und Systeme	
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	60861	3D Scanner - Algorithms and Systems (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0	
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 42910 Advanced Business Process Management

2. Modulkürzel:	052010007	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	052010006 - Business Process Management		
12. Lernziele:	Am Ende der Veranstaltungen haben die Teilnehmer weiterführende Ansätze zur Modellierung von Prozessen und zur Spezifikation von Workflows verstanden. Die Rolle von Muster in der Beschreibung von Workflows ist klar geworden. Verfahren des Process Mining sind theoretisch dargestellt. Die Notwendigkeit zur P2P-Verzahnung ("Choreographien") von Prozessen und entsprechende Ansätze sind klar. Ebenso verstanden ist das darüber hinausgehende Konzept der Komponentenverdrahtung. Weitere Architekturen und Einsatzgebiete von WFMS sind verstanden.		
13. Inhalt:	In der Vorlesung werden fortgeschrittene Themen des Workflowmanagement vorgestellt. Aktuelle Entwicklungen aus dem Forschungsumfeld und der Industrie auf dem Gebiet werden diskutiert. Human Task Management Weitere Ansätze zur Prozessmodellierung (Pi-Kalkül, WSFL, XLANG,...) Muster (Kontrol-, Datenfluss, Organisatorisch) Process Monitoring Process Mining Peer-to-Peer Verzahnung von Prozessen (Choreographie, Gebrauchsanweisungen,...) Verdrahten von Komponenten (Global Models,...) Anwendungsbereiche (Manufakturing, Compliance,...) Prozessadaption und -flexibilität		
14. Literatur:	W. van der Aalst, K. van Hee, Workflow Management, 2002		
15. Lehrveranstaltungen und -formen:			
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	42911 Advanced Business Process Management (PL), schriftlich oder mündlich, Gewichtung: 1.0, schriftlich (60 min) oder mündlich (30 min)		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von: Institut für Architektur von Anwendungssystemen

Modul: 55600 Advanced Information Management

2. Modulkürzel:	051200099	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	PD Holger Schwarz		
9. Dozenten:	• Holger Schwarz • Bernhard Mitschang		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Lecture "Modellierung" or comparable course		
12. Lernziele:	The students learn current concepts for modeling, developing, deploying and processing database-oriented applications. This includes technologies and standards for XML processing and their integration into database systems as well as concepts and systems for content management and data management in the cloud.		
13. Inhalt:	Among the topics to be discussed in this course are: • XML and database technology (XML modeling, XML storage, XML query languages, XML processing) • NoSQL data management (Key value stores, MapReduce, triple stores, document stores, graph stores) • Content management (Enterprise content management, information retrieval, search technologies)		
14. Literatur:	Will be announced at the beginning of the lecture.		
15. Lehrveranstaltungen und -formen:	• 556001 Vorlesung Advanced Information Management • 556002 Übung Advanced Information Management		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	• 55601 Advanced Information Management (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, written (90 min) or oral (20 min), schriftlich (90 min) oder mündlich (20 min) • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 90 Min.		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Datenbanken und Informationssysteme		

Modul: 55740 Advanced Service Computing

2. Modulkürzel:	052010005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	5.0	7. Sprache:	Englisch
8. Modulverantwortlicher:		Univ.-Prof. Frank Leymann	
9. Dozenten:		Frank Leymann	
10. Zuordnung zum Curriculum in diesem Studiengang:		B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module	
11. Empfohlene Voraussetzungen:		Service Computing, Lecture and Exercise (4 SWS) or Services and Service Composition, Lecture and Exercise (4SWS)	
12. Lernziele:		This module comprises two lectures and therefore topics from two areas of advanced service computing. The focus of the Lecture Advanced Service Computing is concepts and technologies for describing and providing stateful resources as Web Services as well as the use of Semantics in Web Services and service compositions. The focus in the Lecture Services and Security is on security aspects of service-based applications.	
13. Inhalt:		This module comprises two lectures and therefore topics from two areas of advanced service computing. Based on the topics discussed in the lecture Service Computing, in the Lecture Advanced Service Computing we will focus on concepts and technologies for describing and providing stateful resources as Web Services. In this respect we will also consider Grid Services and infrastructures. In addition, the topics Semantic Web, Ontologies and Semantic Web Services will be presented in detail. Particular attention will be paid to Semantic Web Service Technologies and frameworks like OWL-S, WSMO, SAWSDL and approaches for their use in service compositions. The focus in the Lecture Services and Security is on security aspects of service-based applications. Foundations of Security in enterprise architectures will be presented, as well as best practices for enterprise and IT security in terms of patterns. Basic Security approaches (e.g. prevention, detection, reaction) and mechanisms (access control, authentication, identification, cryptography) will be presented in detail. We will also discuss current state of the art of Web application and Web Service security.	
14. Literatur:		• Literatur, die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben. • S. Graham, D. Davis, S. Simeonov, G. Daniels, P. Brittenham, Y. Nakamura, P. Fremantle, D. König, C., Building Web Services with Java (2nd Edition), 2005 • S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, Web Services Platform Architecture, 2005	

- Markus Schumacher et al.: Security Patterns: Integrating Security and Systems Engineering, Wiley Series in Software Design Patterns, 2004
- Dieter Gollman: Computer Security, John Wiley & Sons; 3rd Edition, 2010

15. Lehrveranstaltungen und -formen:	• 557401 Advanced Service Computing Lecture (Summer) • 557402 Lecture Services and Security (Winter)
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 60 Stunden Selbststudium: 120 Stunden
17. Prüfungsnummer/n und -name:	55741 Advanced Service Computing (PL), schriftlich oder mündlich, Gewichtung: 1.0, schriftlich (60 min) oder mündlich (30 min)
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Architektur von Anwendungssystemen

Modul: 29550 Algorithmische Geometrie

2. Modulkürzel:	050410105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Funke		
9. Dozenten:	Stefan Funke		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Grundkenntnisse in Datenstrukturen und Algorithmen wie sie in "Datenstrukturen und Algorithmen" (Modul 12060), "Algorithmen und Berechenbarkeit" (Modul 11890), und "Algorithmik" (Modul 10020) vermittelt werden.		
12. Lernziele:	Die Teilnehmer kennen die Grundbegriffe der Algorithmischen Geometrie und haben einen Überblick über die Methoden und Techniken, die in der Algorithmischen Geometrie angewandt werden.		
13. Inhalt:	Es werden die grundlegenden Techniken und Methoden der Algorithmischen Geometrie vermittelt.		
14. Literatur:	de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational Geometry - Algorithms and Applications, Springer, 2000.		
15. Lehrveranstaltungen und -formen:	295501 Vorlesung Algorithmische Geometrie		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 h Selbststudiums- / 138 h Nachbearbeitungszeit: Summe: 180 h		
17. Prüfungsnummer/n und -name:	29551 Algorithmische Geometrie (PL), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Algorithmik		

Modul: 29760 Algorithmische Gruppentheorie

2. Modulkürzel:	050420115	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	Volker Diekert		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Elementare Gruppentheorie		
12. Lernziele:	Die Studierenden beherrschen typische Denk- und Herangehensweisen aus der algorithmischen und kombinatorischen Gruppentheorie. Sie wissen, wie man diverse algorithmische Probleme in freien Gruppen mit Hilfe der Stallingsgraphen lösen kann. Sie können mit Darstellungen von Gruppen durch Erzeugende und Relationen umgehen. Sie kennen das Wortproblem und deren Lösung für gewisse Klassen von Gruppen. Sie kennen konfluente Ersetzungssysteme, HNN-Erweiterungen, amalgamierte Produkte und die Grundbegriffe der Bass-Serre-Theorie.		
13. Inhalt:	Bereits 1911 formulierte Max Dehn drei fundamentale algorithmische Probleme für endlich dargestellte Gruppen: 1) Ist ein gegebenes Gruppenelement g (als Wort in Erzeugern) das Einselement in der Gruppe G? 2) Sind zwei Elemente g und h konjugiert? 3) Definieren zwei gegebene Darstellungen isomorphe Gruppen? Im Allgemeinen sind alle diese Fragen unentscheidbar, also kann man positive Antworten nur in Spezialfällen erhalten. Bei der Lösung des Wortproblems und bei Strukturaussagen ist vor allem die Technik der konfluenten Wortersetzungssysteme hilfreich, die auch in anderen Bereichen zum Einsatz kommen. Insgesamt lebt die Theorie von Querbezügen zu anderen Bereichen, wie Kombinatorik, Topologie, Geometrie, theoretischer Informatik. Dieses Zusammenspiel verschiedener Methoden macht die algorithmische Gruppentheorie sehr attraktiv.		
14. Literatur:	• Björner, Brenti: Combinatorics of Coxeter groups, Springer, 2005. • Camps, Große Rebel, Rosenberger: Einführung in die kombinatorische und geometrische Gruppentheorie, Heidemann Verlag 2008. • Lyndon, Schupp: Combinatorial Group Theory, Springer, 1977. • Magnus, Karrass, Solitar: Combinatorial Group Theory, Wiley & Sons, 1966. • Serre: Trees, Springer, 1980. • Stillwell: Classical Topology and Combinatorial Group Theory, Springer, 1993.		

15. Lehrveranstaltungen und -formen:	297601 Vorlesung mit Übung Algorithmische Gruppentheorie	
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
17. Prüfungsnummer/n und -name:	29761 Algorithmische Gruppentheorie (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0	
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:	Theoretische Informatik	

Modul: 45760 Ausgewählte Kapitel der Algorithmentheorie

2. Modulkürzel:	050410115	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	• Volker Diekert • Stefan Funke • Ulrich Hertrampf		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Grundkenntnisse in Algorithmen und Komplexität		
12. Lernziele:	Die Teilnehmer lernen aktuellste Resultate aus der Algorithmentheorie kennen.		
13. Inhalt:	Es werden aktuelle Forschungsergebnisse in der Algorithmentheorie präsentiert.		
14. Literatur:	aktuelle wissenschaftliche Originalartikel		
15. Lehrveranstaltungen und -formen:	457601 Vorlesung Ausgewählte Kapitel der Algorithmentheorie		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 h Selbststudiums- / 138 h Nachbearbeitungszeit: Summe: 180 h		
17. Prüfungsnummer/n und -name:	45761 Ausgewählte Kapitel der Algorithmentheorie (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Formale Methoden der Informatik		

Modul: 42480 Ausgewählte Kapitel des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240030	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	• Dirk Pflüger • Stefan Zimmer • Miriam Mehl		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	• Modul 10190 Mathematik für Informatiker und Softwaretechniker und • Modul 10240 Numerische und Stochastische Grundlagen der Informatik bzw. • Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker • Modul 42410 Grundlagen des wissenschaftlichen Rechnens		
12. Lernziele:	Die Teilnehmer kennen ausgewählte aktuelle Forschungsthemen des wissenschaftlichen Rechnens und können mit der zugehörigen Primärliteratur arbeiten.		
13. Inhalt:	Aktuelle weiterführende Forschungsthemen des wissenschaftlichen Rechnens, wie z.B. adaptive Finite Elemente, hierarchische Basen und dünne Gitter, robuste Multilevellöser, Wavelets und schnelle Wavelettransformation, p-Version oder Spektralverfahren.		
14. Literatur:	Primärliteratur zu den behandelten Themen: • Bungartz/Griebel: Sparse Grids; Acta Numerica, Volume 13, p. 147-269 • Quarteroni/Valli: Numerical approximation of partial differential equations • Quarteroni: Numerical models for differential problems		
15. Lehrveranstaltungen und -formen:	• 424801 Vorlesung Ausgewählte Kapitel des Wissenschaftlichen Rechnens • 424802 Übung Ausgewählte Kapitel des Wissenschaftlichen Rechnens		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	42481 Ausgewählte Kapitel des Wissenschaftlichen Rechnens (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von: Simulation großer Systeme

Modul: 56680 Automaten über unendlichen Objekten

2. Modulkürzel:	050420230	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	Volker Diekert		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Grundkenntnisse in Mathematik und Theoretischer Informatik. (reguläre Sprachen und endliche Automaten).		
12. Lernziele:	Die Studierenden lernen die wichtigsten Grundtechniken in dem Bereich der formalen Verifikation für nicht terminierende Systeme und nebenläufige Prozess kennen. Sie lernen Denkweisen und Resultate aus verschiedenen mathematischen Disziplinen wie der Topologie, der Logik, oder der Kombinatorik kennen. Sie kennen den Begriff der MSO-Logik und ihre Entscheidbarkeit nach Büchi und Rabin.		
13. Inhalt:	Die Vorlesung behandelt eine mathematischen Theorie für nicht terminierende Systeme und nebenläufige Prozess. Bei der formalen Verifikation kommen Automatenmodelle zum Einsatz, welche unendliche Objekte als Eingabe erhalten. So lassen sich viele Methoden von endlichen Wörtern auf weitere Bereiche wie unendliche Sequenzen oder Bäume ausdehnen. In diesem Sinne ist die Automatentheorie über unendlichen Objekten wesentlich reichhaltiger und spannender als über endlichen Wörtern. Die Vorlesung orientiert sich an den folgenden Themen: • Presburger Arithmetik: Anforderungen an Automaten • Büchi Automaten und omega-reguläre Sprachen • Klarlunds Konstruktion zur Komplementierung von Büchi Automaten • Andere Akzeptanzbedingungen für omega-Automaten • Monadische Logik zweiter Stufe (MSO) • Deterministische omega-Sprachen • Topologisch definierte Sprachklassen • McNaughtons Theorem • Die Safra-Konstruktion • Algebraische Beschreibungen • Eindeutige Büchi Automaten • Logik erster Stufe und andere Fragmente von MSO • Paritätsspiele • Automaten über unendlichen Bäumen • Rabins Baumtheorem		
14. Literatur:	• Volker Diekert, Manfred Kufleitner, Gerhard Rosenberger: Diskrete algebraische Methoden: Arithmetik, Kryptographie, Automaten und Gruppen. De Gruyter, Berlin 2013.		

- Volker Diekert und Paul Gastin: First-order definable languages. In Jörg Flum, Erich Grädel, Thomas Wilke (eds.). Logic and Automata: History and Perspectives. Texts in Logic and Games 2, Amsterdam University Press 2008, pp. 261-306.
- Wolfgang Thomas: Automata on infinite objects. In Jan van Leeuwen (ed.). Handbook of Theoretical Computer Science, volume B: Formal Models and Semantics. Elsevier, 1990, pp. 133-192.
- Wolfgang Thomas: Languages, Automata, and Logic. In Grzegorz Rozenberg and Arto Salomaa (eds). Handbook of Formal Languages, volume 3: Beyond Words. Springer, New York, 1997, pp. 389-455.

15. Lehrveranstaltungen und -formen:	566801	Vorlesung Automaten über unendlichen Objekten
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
17. Prüfungsnummer/n und -name:	56681	Automaten über unendlichen Objekten (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:	Institut für Formale Methoden der Informatik	

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:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	• Thomas Ertl • Daniel Weiskopf		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	• Modul 10060 Computergraphik		
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. Darüber hinaus kennen sie interaktive Verfahren, die unter Ausnutzung programmierbarer Grafik-Hardware realistische Beleuchtungseffekte in Echtzeit approximieren können, sowie bildbasierte Ansätze, die ohne geometrische Daten realistische Darstellungen erzeugen. 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 • Bild-basiertes Rendering		
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 h
Selbststudiums- /	138 h
Nachbearbeitungszeit:	
Summe:	180 h

17. Prüfungsnummer/n und -name:

- 10041 Bildsynthese (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0
- V Vorleistung (USL-V), schriftlich, eventuell mündlich, Übungsschein.

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

Modul: 42900 Business Process Management

2. Modulkürzel:	052010006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	611 Grundlagen der Architektur von Anwendungssystemen, Vorlesung mit Übung, 4,0 SWS		
12. Lernziele:	The course has the objective to provide knowledge about the essential modelling constructs for workflows and their mapping to corresponding workflow languages. In addition, the life cycle of Workflow-based applications will be presented in detail and connected to the Architecture of Workflow Management Systems, which will also be presented. Moreover, the goal is to enable students to use workflow languages (in particular BPEL) in practice. In this respects students will also understand the fundamental approach process graphs, which is applied in workflow languages. Of great importance are , mechanisms for fault handling and exception handling - these will be explained in detail and students will be able to apply them.		
13. Inhalt:	Workflows are IT realisations of business processes and are also considered an approach of significant importance for composition of applications. This course will introduce the foundations of this area, also known as Business Process Management BPM). 1. Historical Development of the Workflow Technology 2. Business Re-engineering (BPM Lifecycle, Tools,...) 3. Architecture of WFMS (Navigator, Executor, Worklist Manager,...) 4. Flow Languages (FDL, BPEL) 5. Process Model Graph (mathematical meta-model: syntax, operational semantics) 6. Advanced functions (sub-processes, event handling, instance modifications, adaptation) 7. Two-level programming paradigm 8. Transactional support in workflows		
14. Literatur:	• F. Leymann, D. Roller, Production Workflow, 2000 • W. van der Aalst, K. van Hee, Workflow Management, 2002		
15. Lehrveranstaltungen und -formen:	429001 Vorlesung mit Übungen, Workflow Management 1		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		

17. Prüfungsnummer/n und -name: 42901 Business Process Management (PL), schriftlich oder mündlich, 60 Min., Gewichtung: 1.0, schriftlich (60 min) oder mündlich (30 min)

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Architektur von Anwendungssystemen

Modul: 57050 Compilerbau

2. Modulkürzel:	051010201	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene 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 verstehen die grundlegende Funktionsweise mehrerer Parse-Verfahren und kennen deren grammatikalischen Einschränkungen. Sie kennen elementare Verfahren semantischer Analysen und sind in der Lage, einfache semantische Prüfungen zu verfassen. Sie haben gelernt, die Fehlermeldungen aus Parser-Generatoren, Compilern oder Interpretern richtig einzuordnen. Ferner haben sie durch Betrachtung der Implementierungsmodelle typischer Programmiersprachenkonstrukte Verständnis für das Ausführungsverhalten erlangt. Sie kennen elementare Begriffe der Codegenerierung und die Eigenschaften von typischen Zwischencodedarstellungen in Compilern.		
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. Attributgrammatiken. Zwischencodeerzeugung. Realisierung einiger Aspekte der Laufzeitsemantik prozeduraler Programmiersprachen. Einfache Codegenerierung.		
14. Literatur:	• A. V. Aho, M.S. Lam, R. Sethi, J. D. Ullman: Compilers - Principles, Techniques, and Tools, Addison Wesley Verlag (2007) • Niklaus Wirth: Compilerbau: Eine Einführung, Teubner Verlag (1986) • Wilhelm, Maurer: Übersetzerbau, 2. Auflage, Springer Verlag (1997) • Andrew W. Appel: Modern Compiler Implementation In Java, Cambridge University Press (2002) • Uwe Kastens: Übersetzerbau, Oldenbourg Verlag (1990)		

15. Lehrveranstaltungen und -formen:	• 570501 Vorlesung Compilerbau • 570502 Übung Compilerbau	
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
17. Prüfungsnummer/n und -name:	57051 Compilerbau (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0	
18. Grundlage für ... :	29660 Programmanalysen und Compilerbau	
19. Medienform:		
20. Angeboten von:	Programmiersprachen und ihre Übersetzer	

Modul: 29570 Computer Interface Technologien

2. Modulkürzel:	051230105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Kenntnisse in mindestens einem Fach der Technischen 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 Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	29571 Computer Interface Technologien (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Minuten oder mündliche Prüfung von 30 Min.		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 29430 Computer Vision

2. Modulkürzel:	051900215	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Andrés Bruhn		
9. Dozenten:	Andrés Bruhn		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	• Modul 10190 Mathematik für Informatiker und Softwaretechniker • Modul 10170 Imaging Science		
12. Lernziele:	Der Student / die Studentin beherrscht die Grundlagen der Merkmalsextraktion und -repräsentation, des 3-D Maschinensehens, der Bildsegmentierung sowie der Mustererkennung. Er/sie kann Probleme aus dem Fachgebiet einordnen und diese selbständig mit den erlernten Algorithmen und Verfahren lösen. The student knows the basics of feature extraction and representation, 3-D computer vision, image segmentation and pattern recognition. He/she can solve problems of the field using the methods discussed in the course.		
13. Inhalt:	• Lineare Diffusion, Skalenräume • Bildpyramiden, Kanten und Eckendetektion • Hough-Transformation, Invarianten • Texturanalyse • Scale Invariant Feature Transform (SIFT) • Bildfolgenanalyse: lokale Verfahren • Bewegungsmodelle, Objektverfolgung, Feature Matching • Bildfolgenanalyse: globale Verfahren • Kamerageometrie, Epipolargeometrie • Stereo Matching und 3-D Rekonstruktion • Shape-from-Shading • Isotrope und anisotrope nichtlineare Diffusion • Segmentierung mit globalen Verfahren • Kontinuierliche Morphologie, Schockfilter • Mean Curvature Motion • Self-Snakes, Aktive Konturen • Bayes'sche Entscheidungstheorie der Mustererkennung • Klassifikation mit parametrischen Verfahren, Dichteschätzung • Klassifikation mit nicht-parametrischen Verfahren • Dimensionsreduktion • Linear Diffusion, Scale Space • Image Pyramids, Edges and Corners • Hough Transform, Invariants • Texture Analysis		

	• Scale Invariant Feature Transform • Image Sequence Analysis: Local Methods • Motion Models, Tracking, Feature Matching • Image Sequence Analysis: Variational Methods • Camera Geometry, Epipolar Geometry • Stereo Matching and 3-D Reconstruction • Shape-from-Shading • Isotropic and Anisotropic Nonlinear Diffusion • Segmentation with Global Methods • Continuous Scaled Morphology, Shock Filters • Mean Curvature Motion • Self-Snakes, Active Contours • Bayes Decision Theory for Pattern Recognition • Classification with Parametric Techniques, Density Estimation • Classification with Non-Parametric Techniques • Dimensionality Reduction	
14. Literatur:	• Forsyth, David and Ponce, Jean: Computer Vision. A Modern Approach, 2003 • Bigun, J.: Vision with Direction, 2006 • L. G. Shapiro, G. C. Stockman: Computer Vision, 2001 • O. Faugeras, Q.-T. Luong: The Geometry of Multiple Images, 2001.	
15. Lehrveranstaltungen und -formen:	• 294301 Vorlesung Computer Vision • 294302 Übung Computer Vision	
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
17. Prüfungsnummer/n und -name:	• 29431 Computer Vision (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben • V Vorleistung (USL-V), schriftlich, eventuell mündlich	
18. Grundlage für ... :	55640 Correspondence Problems in Computer Vision	
19. Medienform:		
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme	

Modul: 29580 Data Compression

2. Modulkürzel:	051230110	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	This course requires basic knowledge in mathematics.		
12. Lernziele:	The students learn the concepts of data compression and acquire an understanding of different algorithms for data compression. Furthermore they will be able to implement and further develop the algorithms discussed in the course.		
13. Inhalt:	Topic of the lecture are algorithms and hardware architectures for data compression • Shannon Entropy • Huffman coding • Universal codes • Arithmetic coding • Lossy and Lossless compression • Image data compression • Dictionary based compression		
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 Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	29581 Data Compression (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, written 90 Min. or oral 30 Min.		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 48480 Data Engineering

2. Modulkürzel:	051210011	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Melanie Herschel		
9. Dozenten:	Melanie Herschel		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 3. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 3. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 3. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Lecture "Modellierung" or comparable course		
12. Lernziele:	The students learn the basic concepts of modeling and system-related issues in data engineering in general and with respect to specific application areas in research-related and engineering-related areas. The methodological basis is defined by information extraction and information analysis, all based on effective metadata management.		
13. Inhalt:	Data engineering involves any data processing necessary to prepare data for subsequent use, e.g., for data analytics or learning. This lecture covers foundations, algorithms, and systems on selected topics of data engineering. These include: • Data collection: how do we find relevant data sources? • Big Data integration: Given the unique properties of big data, how can data from multiple data sources be combined to get a more global perspective on a subject to be analyzed? • Data cleaning: How can important properties and errors of data be assessed and corrected? • Data distribution: What modern technologies support the wide dissemination of data? • Provenance: How can the whole data engineering process be documented, controlled, and improved leveraging so-called provenance meta-data?		
14. Literatur:	There is no unique book covering all aspects of data engineering. The lecture is however significantly based on selected chapters of the following books. • Xin Luna Dong and Divesh Srivastava. Big Data Integration. Synthesis Lectures on Data Management, Morgan an Claypool, 2015. • Wanfei Fan and Floris Geerts. Foundations of Data Quality Management. Synthesis Lectures on Data Management, Morgan an Claypool, 2012. • AnHai Doan, Alon Halevy, and Zachary Ives. Principles of Data Integration. Morgan Kaufmann, 2012. • James Cheney, Laura Chiticariu, and Wang Chiew Tan. Provenance in Databases: Why, How, and Where. Foundations and Trends in Databases, Vol. 1, No.4, 2007.		

15. Lehrveranstaltungen und -formen:	• 484801 Lecture Data Engineering • 484802 Exersice Data Engineering
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden
17. Prüfungsnummer/n und -name:	48481 Data Engineering (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Data Engineering

Modul: 55620 Data Warehousing, Data Mining, and OLAP

2. Modulkürzel:	051210105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Bernhard Mitschang		
9. Dozenten:	• Bernhard Mitschang • Holger Schwarz		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Lecture "Modellierung" or comparable course		
12. Lernziele:	After attending this lecture, students understand the challenges behind the integration of heterogeneous data sources in consolidated warehouses and the provisioning of analytical services. They know the typical data warehouse architecture as well as current trends, e.g., real-time data warehousing. Further topics are the structure of a data warehouse and the main processes for building data warehouses (extraction, transformation, load). A special focus is on technologies to analyze data warehouse data, e.g., reporting, online analytic processing and data mining, and their role as part of analytical services.		
13. Inhalt:	Among the topics to be discussed in this course are: - Introduction to data warehousing - Data warehouse architecture - Data warehouse design - Extraction, transformation, load - ETL as a service - Introduction to analytics and analytic services - Real-time reporting - Online analytic processing - Data mining		
14. Literatur:	• A. Kemper, A. Eickler, Datenbanksysteme - Eine Einführung, 2004 • H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems. The Complete Book, 2003 Will be announced at the beginning of the lecture		
15. Lehrveranstaltungen und -formen:	• 556201 Vorlesung Data Warehousing, Data Mining und OLAP-Technologien • 556202 Übung Data Warehousing, Data Mining und OLAP-Technologien		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		

17. Prüfungsnummer/n und -name:
- 55621 Data Warehousing, Data Mining, and OLAP (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0,
 - V Vorleistung (USL-V), schriftlich oder mündlich, Übungsleistungen während der Unterrichtsperiode als Voraussetzung für die Zulassung zur Prüfung.
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Datenbanken und Informationssysteme

Modul: 29600 Digital System Design II

2. Modulkürzel:	051230122	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	This lectures requires the knowledge of "System Design I". Alternatively, knowledge of "Technische Informatik" is sufficient to follow the course.		
12. Lernziele:	The students will learn to build and implement a complex digital system by using digitals components on a circuit board, and will acquire an in-depth knowledge for implementing complex digital systems using FPGA's.		
13. Inhalt:	• Presentation of a case study of a digital system • Simulatable specification of the system • Architecture for Implementation using FPGAs • Design and design tools for board integration • Implementation of a digital system • Verification of a digital system		
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:	296001 Vorlesung mit Übung Digital System Design II		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29601 Digital System Design II (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 29590 Digitale Systeme

2. Modulkürzel:	051230120	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene 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 Board 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:	295901 Vorlesung mit Übung Digital System Design I		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	• 29591 Digitale Systeme (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündliche Prüfung von 30 Min. • V Vorleistung (USL-V), schriftlich, eventuell mündlich		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 39250 Distributed Systems I

2. Modulkürzel:	051200015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	Kurt Rothermel		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Programmierung und Software-Entwicklung Datenstrukturen und Algorithmen Systemkonzepte und -Programmierung		
12. Lernziele:	The Students will gain an understanding of the basic characteristics, concepts and methods of distributed systems. Furthermore, the ability to analyze existing distributed applications and platforms with regard to its specific properties will be obtained. The implementation of distributed applications as well as system platforms based on the shown methods of that course is another objective. Due to the knowledge provided in that course, the students will be able to communicate with other experts of other professional disciplines, about topics in the field of distributed systems.		
13. Inhalt:	1. Introduction to distributed systems 2. System models 3. Communication: Messages, Remote Procedure Call (RPC), Remote Method Invocation RMI 4. Naming: Generating and Resolution 5. Time Management and clocks in distributed Systems: Applications, logical clocks, physical clocks, synchronization of clocks 6. Global state: concepts, snapshot algorithms, distributed Debugging 7. Transaction management: Serializability, barrier methods, 2-phase-commit-protocols 8. Data replication: primary copy, consensus-protocols and other algorithms 9. Safety/Security: Methods for confidentiality, integrity, authentication and authorization 10. Multicast-algorithms: processing model, broadcast-semantics and algorithms		
14. Literatur:	Literatur, siehe Webseite zur Veranstaltung		
15. Lehrveranstaltungen und -formen:	• 392501 Vorlesung Verteilte Systeme • 392502 Übungen Verteilte Systeme		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 h Selbststudiumszeit / Nachbearbeitungszeit: 138 h Gesamt: 180 h		

17. Prüfungsnummer/n und -name:
- 39251 Distributed Systems I (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Prüfungsdauer: 90 min schriftlich oder 30 min mündlich Exam duration: 90 min written exam or 30 min oral exam
 - V Vorleistung (USL-V), schriftlich, eventuell mündlich
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Verteilte Systeme

Modul: 45730 Distributed Systems II

2. Modulkürzel:	051200169	5. Moduldauer:	2 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	• Kurt Rothermel • Muhammad Tariq		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	The Lecture requires basic knowledge from the course Distributed Systems I		
12. Lernziele:	In this lecture, the acquired knowledge from the previous lecture Distributed Systems I is deepened. The student will gain information about further practice-oriented problems and will implement protocols to solve those problems. The student will be capable to analyze distributed systems in terms of these problems, design, apply and develop protocols for specific applications.		
13. Inhalt:	1. Group communication 2. Consensus 3. Fault tolerant services 4. Wave algorithms 5. Termination 6. Garbage collection 7. Election 8. Deadlocks 9. Organisational & Introduction		
14. Literatur:	• J.L. Welch, H. Attiya, Distributed Computing: Fundamentals, Simulations and Advanced Topics, 1997 The event is based on a collection of scientific papers, which will be announced in the lecture.		
15. Lehrveranstaltungen und -formen:	• 457301 Vorlesung Verteilte Algorithmen • 457302 Vorlesung Asynchronous Middleware Systems		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	45731 Distributed Systems II (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Prüfungsdauer: 90 min schriftlich oder 30 min mündlich Exam duration: 90 min written exam or 30 min oral exam		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Verteilte Systeme

Modul: 57680 Einführung in die Chaostheorie

2. Modulkürzel:	074810350	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Allgöwer		
9. Dozenten:	Viktor Avrutin		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Die Teilnehmer lernen die Grundbegriffe der Theorie der nichtlinearen dynamischen Systeme bzw. der Chaostheorie kennen. Die Studierenden verstehen solche Begriffe wie zeit-kontinuierliche und zeit-diskrete Modellierung, transiente und asymptotische Dynamik, Attraktoren, Stabilität, Bifurkationen, Bifurkationsszenarien, Deterministisches Chaos, "Wege ins Chaos". Sie können verschiedene Typen von lokalen und globalen Bifurkationen erkennen und kennen auch die Bedingungen, die zu diesen Bifurkationen führen. Darüber hinaus lernen die Studierenden die typischen quantitativen Maße kennen, die bei der praktischen Untersuchung des Verhaltens angewendet werden. Dazu zählen in erster Linie Lyapunov-Exponenten, fraktale Dimensionen und Entropien. Ein wesentlicher Teil der Vorlesung ist einem modernen Kapitel der Nichtlinearen Dynamik gewidmet, nämlich der Theorie der stückweise-glatte Systeme. Die Studierenden lernen die für diese Systeme charakteristischen Phänomene (border-collision bifurcations, period-adding) kennen, sowie Konzepte der Symbolischen Dynamik und die typischen Anwendungen aus dem technischen Bereich (impacting systems, switching circuits). Abschließend wird in der Vorlesung der Zusammenhang zwischen dynamischen Systemen und Fraktalen gezeigt. Die Studierenden verstehen darauf die Bedeutung der Standard-Beispiele aus diesem Gebiet (Cantor-Mengen, Julia-Mengen, Mandelbrot-Mengen). Ein besonderer Wert wird in dieser Lehrveranstaltung darauf gelegt, dass die Teilnehmer eigene praktische Erfahrungen im Umgang mit dynamischen Systemen (am Beispiel von niedrig-dimensionalen zeit-diskreten Abbildungen) sammeln. Zu diesem Zweck bietet die Vorlesung den Studierenden die Möglichkeit, viel zu experimentieren.		
13. Inhalt:	1. Problemstellungen und Grundbegriffe 2. Qualitative Analyse: Attraktoren (periodische, aperiodische, chaotische Trajektorien), Bifurkationen (lokale und globale Bifurkationen, Bifurkationen in stückweise-glatte Systemen); Bifurkations-szenarien (in glatten und stückweise-glatte Systemen) 3. Quantitative Analyse: Lyapunov Exponenten, fraktale Dimensionen, weitere Maße. Symbolische Dynamik		

4. Fraktale

14. Literatur:	John Argyris, Gunter Faust, Maria Haase, Rudolf Friedrich , Die Erforschung des Chaos: Eine Einführung in die Theorie nichtlinearer Systeme (Springer, 2010)
	Skript
15. Lehrveranstaltungen und -formen:	576801 Vorlesung Einführung in die Chaostheorie
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42; Selbststudium: 138
17. Prüfungsnummer/n und -name:	57681 Einführung in die Chaostheorie (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

Modul: 29710 Embedded Systems Engineering

2. Modulkürzel:	051711027	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	keine		
12. Lernziele:	Master-level understanding of the design methodology and advanced design techniques for constructing and analyzing embedded hardware / software systems.		
13. Inhalt:	1. Introduction to embedded systems and their design constraints 2. Synthesis models and algorithms 3. System level synthesis 4. High level synthesis 5. Pipelined data path and controller design 6. Software task scheduling and schedulability analysis 7. Static and dynamic methods for scheduling and priority assignment 8. Communication architectures for embedded systems		
14. Literatur:	Skript „Embedded Systems Engineering“ G. Buttazzo: Hard Real Time Computing Systems. 2nd edition, Springer, 2005 P. Eles, K. Kuchcinski, Z. Peng: System Synthesis with VHDL. Kluwer Academic Publishers, 1998. P. Marwedel: Embedded Systems Design. Springer, 2006		
15. Lehrveranstaltungen und -formen:	• 297101 Vorlesung Embedded Systems Engineering • 297102 Übung Embedded Systems Engineering		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Summe: 180 Stunden		
17. Prüfungsnummer/n und -name:	• 29711 Embedded Systems Engineering (Klausur) (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich, Als Zulassungsvoraussetzung zur Klausur ist folgende Vorleistung zu erbringen: Teilnahme an den Übungen, Präsentation der Lösung wenigstens einer Aufgabe.		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von: Eingebettete Systeme (Embedded Systems Engineering)

Modul: 58190 Entwurf und Implementierung eines Compilers

2. Modulkürzel:	05151313	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	• Erhard Plödereder • Timm Felden		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Modul Compilerbau ist notwendige Voraussetzung, Java-Kenntnisse werden erwartet. Die Teilnehmerzahl in diesem Modul ist auf maximal 15 beschränkt.		
12. Lernziele:	Die Studierenden haben praktische Erfahrungen mit der Konstruktion eines Compilers und der Umsetzung von Konzepten in Programmiersprachen erworben. Sie sind in der Lage aktuelle Entwicklungen im Bereich der Programmiersprachen und des Compilerbaus zu beurteilen. Durch die Teilnahme an Programmierübungen mit Codereviews haben sie gelernt, qualitativ hochwertige Compiler zu entwickeln.		
13. Inhalt:	Lexer- und Parsergeneratoren, Semantische Attributierung, Fehlererkennung und -behandlung in Compilern, Typsysteme und Typprüfung, Die Java Virtual Machine, Zwischencodengenerierung, Sprachinterfaces		
14. Literatur:	A.W. Appel : Modern Compiler Implementation in Java 2nd Edition; Cambridge University Press (2002) A. V. Aho, M.S. Lahm, R. Sethi, J. D. Ullman: Compilers - Principles, Techniques, and Tools; Addison, Wesley (2007)		
15. Lehrveranstaltungen und -formen:	581901 Vorlesung Entwurf und Implementierung eines Compilers		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	• 58191 Entwurf und Implementierung eines Compilers (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 56980 Fachpraktikum Cloud Architekturen und Management

2. Modulkürzel:	[pord.modulcode]	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Johannes Wettinger		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Obligatory: Object-oriented programming (e.g. Java), XML Optional: scripting languages (e.g. Python, Ruby, Bash), JSON, YAML		
12. Lernziele:	The goal of this course is twofold: First, established reference architectures are discussed and used to model scalable architectures for Cloud applications. Second, based on these architectures, different Cloud management approaches are used to provision infrastructure resources, deploy application components, and manage them. The course is targeted at students of MSc Computer Science, MSc Softwaretechnik, and MSc Informatik.		
13. Inhalt:	Architectures for scalable, elastic, and reliable Cloud applications Cloud infrastructure management (OpenStack, Amazon Web Services) Configuration management (Chef, Puppet, CFEngine, etc.) Container virtualization (Docker, LXC, etc.) PaaS-centric management (Cloud Foundry, BOSH, etc.) Model-driven Cloud management: infrastructure-centric models (CloudFormation, Heat, etc.), application-centric models (Juju, TOSCA, etc.)		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	569801 Fachpraktikum Cloud Architekturen und Management		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	56981 Fachpraktikum Cloud Architekturen und Management (LBP), Studienbegleitend, 0 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 29740 Fachpraktikum Eingebettete Systeme

2. Modulkürzel:	051711135	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	6.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	wiss. MA		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Modul "Embedded Systems Engineering"		
12. Lernziele:	Ability to model, simulate, and synthesize digital circuits and systems using state-of-the-art description languages and modelling styles for various abstraction levels.		
13. Inhalt:	1) Introduction to VHDL 2) Dataflow modelling 3) Behavioural modelling 4) Structural modelling 5) VHDL library concept 6) RTL modelling 7) Synthesis 8) Introduction to SystemC 9) Hierarchy and basic communication 10) Channels, events, time, and simulation mechanisms 11) Advanced communication modelling 12) Transaction level modelling (TLM) 13) The OSCI TLM 2.0 standard		
14. Literatur:	Lab handouts Documentation of development tools (provided in the lab) Peter Ashenden: The Designer's Guide to VHDL (book available in the lab) Black et al.: SystemC from the Ground Up (book available in the lab)		
15. Lehrveranstaltungen und -formen:	297401 Übung Fachpraktikum Eingebettete Systeme		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit im Labor: 64 Stunden Vor- und Nachbereitung: 116 Stunden Summe: 180 Stunden		
17. Prüfungsnummer/n und -name:	29741 Fachpraktikum Eingebettete Systeme (LBP), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von: Eingebettete Systeme (Embedded Systems Engineering)

Modul: 29750 Fachpraktikum Rechnerarchitektur

2. Modulkürzel:	051700025	5. Moduldauer:	1 Semester								
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe								
4. SWS:	4.0	7. Sprache:	Englisch								
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich										
9. Dozenten:	• Hans-Joachim Wunderlich • Chang Liu • Laura Rodriguez Gomez										
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module										
11. Empfohlene Voraussetzungen:	• Modul 41930 Rechnerorganisation • Modul 10140 Advanced Processor Architecture										
12. Lernziele:	Students are able to design digital sytems by using the complete state of the art design automation tool chain.										
13. Inhalt:	In this lab course, the students design and implement a RISC processor and extend it with techniques common for high-performance processors. Hardware structures found in the state of the art processors will be applied and adapted. In order to achieve high frequency, proper design and verification techniques play an important role. The students learn how timing analysis, pipelining and retiming can be used to optimize the synthesis results. Because software has to be specifically tailored to such a processor architecture, the lab course also deals with scheduling techniques that allow to avoid pipeline stalls and hazards.										
14. Literatur:	• D. A. Patterson, J. L. Hennessy: Computer Organization & Design . The Hardware / Software Interface (3rd Edition); San Francisco, Ca.: Morgan Kaufmann Publishers Inc., 2004 • J. L. Hennessy and D. A. Patterson: Computer Architecture - A Quantitative Approach (3rd Edition); San Francisco, Ca.: Morgan Kaufmann Publishers Inc., 2012										
15. Lehrveranstaltungen und -formen:	297501 Fachpraktikum Rechnerarchitektur										
16. Abschätzung Arbeitsaufwand:	<table><tr><td>Präsenzzeit:</td><td>42 h</td></tr><tr><td>Selbststudiums- /</td><td>138 h</td></tr><tr><td>Nachbearbeitungszeit:</td><td></td></tr><tr><td>Summe:</td><td>180 h</td></tr></table>			Präsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Präsenzzeit:	42 h										
Selbststudiums- /	138 h										
Nachbearbeitungszeit:											
Summe:	180 h										
17. Prüfungsnummer/n und -name:	29751 Fachpraktikum Rechnerarchitektur (LBP), schriftlich oder mündlich, Gewichtung: 1.0										
18. Grundlage für ... :											
19. Medienform:											

20. Angeboten von: Rechnerarchitektur

Modul: 24900 Fachpraktikum Rechnergestützte Methoden der Produktentwicklung

2. Modulkürzel:	051400006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dieter Roller		
9. Dozenten:	• Dieter Roller • Akram Chamakh • Julian Eichhoff		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 4. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Grundkenntnisse wie sie in „CAD und Produktmodelle“ vermittelt werden.		
12. Lernziele:	Die Studierenden sind nach erfolgreichem Besuch des Moduls in der Lage • typische Problemstellungen, Methoden und Technologien im Anwendungsbereich des jeweils behandelten CAx-Feldes sowie dessen Eingliederung in die Produktentwicklungskette unter Verwendung von Fachvokabular zu beschreiben • die vorgestellten Methoden und Technologien gegenüberzustellen und ihren Einsatz im Bezug zu vorgegebenen Problemstellungen zu begründen • die Funktionen eines Technologievertreters aus dem CAx-Feld bei einer vorgegebenen Problemstellung effektiv anwenden können		
13. Inhalt:	Jedes Semester wechselnd werden zu einem konkreten CAx-Bereich (CAD, CAM, CAP, CAQ) folgende Inhalte behandelt: • Eingliederung des Unternehmensbereichs hinsichtlich Produktentstehung/Produktlebenszyklus; Funktionen, Prozesse und Tätigkeiten des Unternehmensbereichs; Informationsflüsse (zugrundeliegende Daten, Objekte und Artefakte) und weitere Verbindung zu anderen Unternehmensbereichen • Überblick zu typischen Problemstellungen und darauf anwendbare Methoden im Aufgabenfeld des Unternehmensbereichs; Methodenvergleich • Überblick der marktdominierenden CAx-Technologien zur Unterstützung des Aufgabenfeldes; Technologievergleich • Anwendungsbezogene Detailvorstellung der Funktionen und Repräsentationsformen eines Vertreters dieser Technologien im Rahmen konkreter Beispielpunkte		
14. Literatur:	• D. Roller. CAD: Effiziente Anpassungs- und Variantenkonstruktion. Springer, Heidelberg, 1995. • S. Vajna, C. Weber, H. Bley, K. Zeman. CAx für Ingenieure - Eine praxisbezogene Einführung. Springer, Heidelberg, 2009.		

- G. Pahl, W. Beitz, J. Feldhusen, K.-H. Grote. Konstruktionslehre - Grundlagen erfolgreicher Produktentwicklung Methoden und Anwendung. Springer, Heidelberg, 2007.

15. Lehrveranstaltungen und -formen:	249001 Übung Fachpraktikum Rechnergestützte Methoden der Produktentwicklung								
16. Abschätzung Arbeitsaufwand:	<table> <tr> <td>Präsenzzeit:</td><td>42 h</td></tr> <tr> <td>Selbststudiums- /</td><td>138 h</td></tr> <tr> <td>Nachbearbeitungszeit:</td><td></td></tr> <tr> <td>Summe:</td><td>180 h</td></tr> </table>	Präsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Präsenzzeit:	42 h								
Selbststudiums- /	138 h								
Nachbearbeitungszeit:									
Summe:	180 h								
17. Prüfungsnummer/n und -name:	24901 Fachpraktikum Rechnergestützte Methoden der Produktentwicklung (LBP), schriftlich oder mündlich, Gewichtung: 1.0								
18. Grundlage für ... :									
19. Medienform:									
20. Angeboten von:	Institut für Rechnergestützte Ingenieursysteme								

Modul: 45750 Fachpraktikum Verteilte Systeme

2. Modulkürzel:	051200111	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	Frank Dürr		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Verteilte Systeme, Rechnernetze II		
12. Lernziele:	Die Teilnehmer besitzen die Fähigkeit, verteilte Anwendungen und Dienste zu entwerfen und zu implementieren. Sie besitzen praktische Kenntnisse in der Netzprogrammierung und der Programmierung von Client/Server-Anwendungen. Sie verfügen über praktische Kenntnisse über Technologien und Werkzeugen zur Implementierung und zum Testen verteilter Systeme.		
13. Inhalt:	- Socket-Programmierung - Höherwertige Kommunikationsprotokolle und Webservice- Schnittstellen (HTTP & XML/JSON, RPC, SOAP, REST) - Client/Server-Systeme - Peer-to-Peer- und Ad-hoc-Kommunikation - Entwicklungsumgebungen - Test verteilter Systeme		
14. Literatur:	- A.S. Tanenbaum: Computer Networks, 4th Edition, 2003		
15. Lehrveranstaltungen und -formen:	457501 Fachpraktikum Verteilte Systeme		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	45751 Fachpraktikum Verteilte Systeme (PL), Sonstiges, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Verteilte Systeme		

Modul: 58440 Fachpraktikum: Algorithmik

2. Modulkürzel:	[pord.modulcode]	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Funke		
9. Dozenten:	Stefan Funke		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:			
12. Lernziele:			
13. Inhalt:			
14. Literatur:	wird in der Veranstaltung bekanntgegeben		
15. Lehrveranstaltungen und -formen:	584401 Fachpraktikum Algorithmik		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 h Selbststudiums- / 138 h Nachbearbeitungszeit: Summe: 180 h		
17. Prüfungsnummer/n und -name:	58441 Fachpraktikum: Algorithmik (LBP), schriftlich und mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Formale Methoden der Informatik		

Modul: 29440 Geometric Modeling and Computer Animation

2. Modulkürzel:	051900010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Daniel Weiskopf		
9. Dozenten:	• Daniel Weiskopf • Thomas Ertl • Guido Reina		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Basic computer graphics, for example: • 10060 Computergraphik		
12. Lernziele:	Students gain an understanding of the fundamental concepts and techniques of geometric modeling and computer animation. This includes theoretical and mathematical foundations, important algorithms, and implementation aspects as well as practical experience with modeling and animation tools such as Maya.		
13. Inhalt:	This course covers foundations and methods for the modeling of scenes and for computer animation. This includes the representation of curves and surfaces, which are used by modeling and animation software for modeling of objects, description of the dynamics of parameters, or keyframe animation. Physically based animation describes motion via kinematic and dynamics laws of mechanics. Applications thereof include particle systems all the way to character animation and deformation. In particular, the following topics are covered: • Description and modeling of curves: differential geometry of curves, polynomial curves in general, interpolation, Bezier curves, B-splines, rational curves, NURBS • Description and modeling of surfaces: differential geometry of surfaces, tensor product surfaces, Bezier patches, NURBS, ruled surfaces, Coons pathes • Subdivision schemes: basic concept, convergence and limit process, subdivision curves, subdivision surfaces • Overview of animation techniques • Keyframe animation, inverse kinematics • Physically based animation of points and rigid bodies: kinematics and dynamics • Particle systems: Reeves, flocking and boids, agent-based simulation • Cloth animation: continuum mechanics, mass-spring model, numerical solvers for ordinary differential equations, explicit and implicit integrators		

	• Collision: efficient collision detection, bounding volume hierachies, hierarchical space partitioning, collision handling, sliding and resting contact • Fluid simulation: wave equation, Navier Stokes, level sets, particle level sets • Basics of film production: camera, lighting, production process, storyboard
14. Literatur:	• D. Eberly, 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics. Morgan Kaufmann, 2000 • G. Farin: Curves and Surfaces for CAGD: A Practical Guide. Morgan Kaufmann, 2002 • R. Parent: Computer Animation: Algorithms and Techniques. Morgan Kaufmann, 2002 • W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling: Numerical Recipies - The Art of Scientific Computing. Cambridge University Press, 1986
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 Geometric Modeling and Computer Animation (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0, • V Vorleistung (USL-V), schriftlich, eventuell mündlich, Erfolgreiche Teilnahme an Übungen / exercises passed
18. Grundlage für ... :	
19. Medienform:	Video projector, blackboard, exercises using PCs
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme

Modul: 29450 Graphentheorie

2. Modulkürzel:	050420105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	• Ulrich Hertrampf • Volker Diekert • Manfred Kufleitner		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Grundvorlesungen in theoretischer Informatik		
12. Lernziele:	Die Studierenden beherrschen typische Denk- und Herangehensweisen aus der Graphentheorie. Die Beziehung zwischen diversen Graphparametern werden verstanden, ebenso wie ihre algorithmische Relevanz. Die Eigenschaften der wichtigsten Graphklassen erschließen sich den Studierenden.		
13. Inhalt:	Die Vorlesung behandelt algorithmische Problem und strukturelle Zusammenhänge bei Graphen. Im Einzelnen werden die folgenden Themen behandelt: • Eulergraphen • Cographen • Bipartite Graphen • Planare Graphen, Eulerformel, Satz von Kuratowski • Graphparameter • Perfekte Graphen • Graphenfärbungen und der Satz von Ramsey • Extremale Graphentheorie		
14. Literatur:	• Reinhard Diestel: Graphentheorie. Springer, 2010. • Martin Aigner, Günter M. Ziegler: Das BUCH der Beweise. Springer, 2009. • Jacobus H. van Lint, Richard M. Wilson: A Course in Combinatorics. Cambridge University Press, 2nd edition, 2001.		
15. Lehrveranstaltungen und -formen:	294501 Vorlesung mit Übungen Graphentheorie		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 h Selbststudiums- / 138 h Nachbearbeitungszeit: Summe: 180 h		
17. Prüfungsnummer/n und -name:	• 29451 Graphentheorie (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich oder mündlich		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Theoretische Informatik

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:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	• Hans-Joachim Wunderlich • Michael Kochte		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	• Modul 10140 Grundlagen der Rechnerarchitektur / Advanced Processor Architecture • Modul 10310 Rechnerorganisation		
12. Lernziele:	• Knowledge of methods for reliability assessment of circuits and systems • Knowledge of the main techniques for implementing fault tolerance • Knowledge how to design fault tolerant circuits and systems		
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 and E. Fujiwara: Error Control Coding for Computer Systems, Prentice Hall (1989) • M.L. Bushnell and 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 h
	Self Study:	138 h
	Sum:	180 h
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	

Modul: 14380 Hardware Verification and Quality Assessment

2. Modulkürzel:	051700020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	• Hans-Joachim Wunderlich • Michael Kochte • Laura Rodriguez Gomez		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 4. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 4. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 4. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	• Modul 10310 Rechnerorganisation oder • Modul 10140 Grundlagen der Rechnerarchitektur		
12. Lernziele:	Basic knowledge of methodologies and algorithms of functional and formal verification, diagnosis, test and design for testability of integrated circuits		
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		
15. Lehrveranstaltungen und -formen:	• 143801 Vorlesung Hardware Verification and Quality Assessment • 143802 Übung Hardware Verification and Quality Assessment		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h

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: Institut für Technische Informatik

Modul: 42920 Hardware-Software-Codesign

2. Modulkürzel:	051711110	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Bachelor-Veranstaltung "Grundlagen der Eingebetteten Systeme" oder gleichwertige Kenntnisse		
12. Lernziele:	Ability to conceptualize systems so that an application-specific, optimized trade-off between hardware and software implementation of system functionality is achieved.		
13. Inhalt:	This module deals with the joint design and optimization of hardware and software for pre-defined applications, covering the following topics: 1. Models for system specification 2. Modelling and simulation with the SystemC library 3. Synthesis of system architectures 4. Resource allocation and operation binding 5. Partitioning of functionality among hardware and software 6. Scheduling and schedulability for parallel multi-core architectures 7. Methods for system optimization 8. Application specific instruction set processors (ASIPs) 9. Network-on-Chip (NoC) interconnect architectures		
14. Literatur:	J. Teich, Digitale Hardware/Software-Systeme, 2. Auflage, 2007		
15. Lehrveranstaltungen und -formen:	• 429201 Vorlesung Hardware-Software-Codesign • 429202 Übung Hardware-Software-Codesign		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	42921 Hardware-Software-Codesign (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)		

Modul: 42420 High Performance Computing

2. Modulkürzel:	051240040	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	• Martin Bernreuther • Dirk Pflüger • Miriam Mehl • Stefan Zimmer		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	• Modul 10190 Mathematik für Informatiker und Softwaretechniker und • Modul 10240 Numerische und Stochastische Grundlagen der Informatik bzw. • Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker		
12. Lernziele:	• Fähigkeit, parallele Algorithmen auf unterschiedlichen parallelen Plattformen mit Hilfe geeigneter algorithmischer Modelle zu bewerten. • Kenntnis verschiedener Programmiermodelle für Parallelrechner mit verteiltem und gemeinsamem Speicher. • Fähigkeit, auch fortgeschrittene Implementierungsaufgaben aus dem Bereich des Höchstleistungsrechnens auf Basis ausgewählter Programmiermodelle zu bewältigen.		
13. Inhalt:	Die Vorlesung beschäftigt sich mit den Grundlagen paralleler Programmierung und paralleler Algorithmen speziell im Hinblick auf die Anwendungsbereiche Wissenschaftliches Rechnen und High Performance Computing. Verwandte Fragestellungen aus dem Bereich der Theorie (parallele Modelle und parallele Komplexität, etc.) sowie aus der Rechnertechnik (parallele Architekturen) werden begleitend diskutiert. Nach einer allgemeinen Einführung (Klassifizierung von Parallelrechnern, Ebenen von Parallelität, Performance und Architekturen, etc.), werden die Grundlagen paralleler Programme eingeführt (Notation/Syntax, Synchronisation und Kommunikation, Design paralleler Programme, etc.). Sowohl die Programmierung auf Systemen mit gemeinsamem Speicher als auch auf Systemen mit verteiltem Speicher werden besprochen. Dabei wird jeweils mindestens ein geeignetes Programmiermodell (z.B. OpenMP, MPI, CUDA) vertieft behandelt. Aus dem Bereich des High Performance Computing werden begleitend klassische Algorithmen und Implementierungstechniken als Beispiele behandelt, z.B. parallele Algorithmen aus der linearen Algebra (Matrixmultiplikation, etc. oder einfache Verfahren zur Lösung partieller Differentialgleichungen). Zusätzlich können Themen wie Lastverteilung		

und Lastbalancierung (Grundlagen, Algorithmen zur Partitionierung und Lastbalancierung, etc.) vorgestellt werden.

14. Literatur:

- T. Rauber, G. Rünger: „Parallele Programmierung“, 2. Aufl., Springer 2007; (in English: T. Rauber, G. Rünger: „Parallel Programming: for Multicore and Cluster Systems“, Springer 2010)
- K.A. Berman, J.L. Paul: "Sequential and Parallel Algorithms", PWS Publishing Company, 1997
- B. Chapman, G. Jost, R. van der Pas: "Using OpenMP - Portable Shared Memory Parallel Programming", MIT Press, 2008
- W. Gropp, E. Lusk, und R. Thakur: "Using MPI-2: Advanced Features of the Message-Passing Interface", das Buch ist auch in deutscher Übersetzung erhältlich.
- D. Kirk, W.-M. Hwu Programming Massively Parallel Processors

15. Lehrveranstaltungen und -formen:

- 424201 Vorlesung High Performance Computing
- 424202 Übung High Performance Computing

16. Abschätzung Arbeitsaufwand:

Präsenzzeit:	42 h
Selbststudiums- /	138 h
Nachbearbeitungszeit:	
Summe:	180 h

17. Prüfungsnummer/n und -name:

42421 High Performance Computing (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Simulation großer Systeme

Modul: 51720 IT-Strategy

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Sven Lorenz		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Die Vorlesung fokussiert auf Management Strategien. Es wird erläutert, wie solche Strategien entwickelt und evaluiert werden. Teilnehmer der Vorlesung verstehen die Bestandteile einer IT Strategie. Sie können eine IT Strategie ableiten und entwickeln, basierend auf dem aktuellen Status eines Unternehmens. Insbesondere wird verstanden, was unter den Begriffen und Konzepten IT Organisation, Sourcing Management, Architektur Management, Qualitäts- und Risk-Management und schliesslich IT Landschaften zu verstehen ist und wie man damit umgeht.		
13. Inhalt:	Über die Einstiegsfragestellung „Was ist ‚Strategie‘?“ wird erläutert, was eine Unternehmensstrategie und eine IT-Strategie ist, wobei sowohl die klassischen Ansätze als auch neue Sichtweisen vorgestellt werden. Im Schwerpunkt „Strategieentwicklung“ wird auf die Ableitung der IT-Strategie aus der Unternehmensstrategie eingegangen. Ein kanonisches Vorgehensmodell wird eingeführt und anhand von Unternehmensbeispielen illustriert. Der Schwerpunkt „IT-Strategie als Prozess“ beginnt mit der Einbettung der IT-Strategieaufgaben in die bekannten IT Prozessmodelle wie ITIL und CobiT. Im Rahmen eines verallgemeinerten IT-Prozessmodells werden die einzelnen IT-Strategieprozesse (IT-Organisationsentwicklung, IT-Sourcing-Strategie, IT-Architektur-Management, IT-Bebauungsplanung, IT-Qualitätsmanagement und IT-Risikomanagement) in der Folge detailliert erläutert. Dabei werden klassische und State-of-the-art Methoden und Werkzeuge zur Unterstützung der IT-Strategieprozesse vorgestellt. Exkurse in das IT-Portfoliomanagement und in IT-Kennzahlensysteme runden die Vorlesungsinhalte ab.		
14. Literatur:	• Helmut Krcmar, „Informationsmanagement“, Springer, 2010 • Jürgen Hofmann, Werner Schmitt, „Masterkurs IT-Management“, VIEWEG+TEUBNER, 2010W. • Brenner, A. Resch, V. Schulz, „Die Zukunft der IT in Unternehmen“, FAZ Buch, 2010 • Martin Kütz, „Kennzahlen in der IT“, dpunkt-Verlag, 2007		
15. Lehrveranstaltungen und -formen:	517201 Vorlesung mit Übungen IT-Strategie		

16. Abschätzung Arbeitsaufwand:	Lecture & exercises: 42 hours Self-study: 138 hours
17. Prüfungsnummer/n und -name:	51721 IT-Strategy (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Architektur von Anwendungssystemen

Modul: 48500 Image Synthesis

2. Modulkürzel:	051903654	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	• Thomas Ertl • Daniel Weiskopf		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Modul 10060 Computergraphik		
12. Lernziele:	The students know the theoretical foundations of image synthesis and have practical expertise in programming of rendering systems. They know several approaches and algorithms for three-dimensional computer graphics, both for real-time and physically accurate rendering.		
13. Inhalt:	The class covers physically based rendering techniques such as ray/path tracing and radiosity, computer graphics models for light transport and light/scene interaction, as well as numerical methods such as Monte Carlo integration and finite element methods which approximate solutions to the rendering equation. In addition, techniques which specifically employ modern graphics processing hardware are covered which approximate physically correct solutions in interactive application scenarios by means of rasterization and image-space rendering. Specifically, the class covers: • graphics hardware and rasterization APIs by example of OpenGL • textures and procedural models • shading and shadow computations in rasterization pipelines • scene graphs, culling and level-of-detail approaches • physically based rendering and photo-realistic image synthesis • local shading and material models, especially the BRDF • the rendering equation • ray tracing and Monte-Carlo approaches • global illumination simulation (especially by means of radiosity, distribution ray tracing and path tracing)		
14. Literatur:	• Andrew S. Glassner: Principles of Digital Image Synthesis, 1995J. • Foley, A. van Dam, S. Feiner, J. Hughes: Computer Graphics: Principle and Practice, 1990M. • Pharr, G. Humphreys: Physically Based Rendering, 2004		
15. Lehrveranstaltungen und -formen:	• 485001 Lecture Image Synthesis • 485002 Exercise Image Synthesis		

16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
17. Prüfungsnummer/n und -name:	• 48501 Image Synthesis (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0	
	• V Vorleistung (USL-V), schriftlich oder mündlich	
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme	

Modul: 55610 Information Integration

2. Modulkürzel:	051210166	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Melanie Herschel		
9. Dozenten:	Melanie Herschel		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Lecture "Modellierung" or comparable course		
12. Lernziele:	Integrating heterogeneous, autonomous and structured data is essential in an interconnected world. This is the basis for information exchange and comprehensive search. The goal of this course is to provide an overview of challenges in information integration and to enable the students to assess available approaches and technologies.		
13. Inhalt:	Based on application scenarios from various organizations, we will discuss aspects of distribution, autonomy and heterogeneity. This helps us to organize the problem space and to compare possible architectures of integrated information systems. Heterogeneity is addressed by schema mappings between and data mappings. We will discuss how to establish such mappings and how to apply them in data transformation. As query processing in federated databases is based on these mappings as well, we will also learn the basics on these systems. Another focus of this course is on the pre-processing and integration of data. Starting with a discussion on information quality, we will look at the spectrum of erroneous data and approaches to data cleansing. State-of-the-art software for information integration will be presented, in particular as part of the exercises.		
14. Literatur:	Additional literature will be announced at the beginning of the lecture • Ulf Leser, Felix Naumann: Informationsintegration: Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen, dpunkt Verlag, 2006, ISBN 3898644006. • AnHai Doan, Alon Halevy, Zachary Ives: Principles of Data Integration, Morgan Kaufmann, 2012, ISBN 978-0-12-416044-6		
15. Lehrveranstaltungen und -formen:	• 556101 Vorlesung Information Integration • 556102 Übung Information Integration		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		

17. Prüfungsnummer/n und -name: 55611 Information Integration (PL), schriftlich oder mündlich, 90 Min.,
Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Data Engineering

Modul: 55630 Information Visualization and Visual Analytics

2. Modulkürzel:	051900099	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Daniel Weiskopf		
9. Dozenten:	• Steffen Koch • Thomas Ertl • Daniel Weiskopf		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Basic Human Computer Interaction		
12. Lernziele:	Student gains expertise about fundamental concepts and techniques of information visualization and visual analytics. This includes algorithms and mathematical background, data structures and implementation aspects as well as practical experience with widely available visualization tools.		
13. Inhalt:	Topics covered in this course: - Perception and Cognition - Graphs and Networks - Hierarchies and Trees - Multi-dimensional and high-dimensional data visualization - Time series visualization - Visual Analytics - Software Visualization - Geospatial visualization		
14. Literatur:	• Colin Ware. Visual Thinking for Design • Colin Ware. Information Visualization. Perception for Design • Edward Tufte. The Visual Display of Quantitative Information • Robert Spence. Design for Interaction • Jim Thomas. Illuminating the Path		
15. Lehrveranstaltungen und -formen:	556301 Vorlesung und Übung Informationsvisualisierung		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		

17. Prüfungsnummer/n und -name:
- 55631 Information Visualization and Visual Analytics (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0
 - V Vorleistung (USL-V), schriftlich, eventuell mündlich, Erfolgreiche Übungsteilnahmen / excercises passed
-

18. Grundlage für ... :

19. Medienform: Video projector, blackboard, exercises using PCs

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

Modul: 60120 Interaktive Systeme

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Albrecht Schmidt		
9. Dozenten:	• Albrecht Schmidt • Niels Henze		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Vorlesung Mensch-Computer Interaktion		
12. Lernziele:	Die Studierenden lernen, wie interaktive Systeme entwickelt werden. Sie verstehen den Entwicklungsprozess und können interaktive Systeme für spezifische Plattform entwickeln.		
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:	601201 Fachpraktikum Interaktive Systeme		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 56 Stunden Selbststudium: 124 Stunden		
17. Prüfungsnummer/n und -name:	60121 Interaktive Systeme (LBP), Sonstiges, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 29460 Kryptographische Verfahren

2. Modulkürzel:	050420110	5. Moduldauer:	1 Semester								
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig								
4. SWS:	4.0	7. Sprache:	Deutsch								
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf										
9. Dozenten:	• Ulrich Hertrampf • Volker Diekert • Stefan Funke										
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module										
11. Empfohlene Voraussetzungen:	Theorie-Vorlesungen des Bachelor-Studiums										
12. Lernziele:	Die Studierenden kennen die wichtigsten Sätze der Kryptographie. Sie können klassische und moderne Verschlüsselungsverfahren anwenden und die Sicherheit dieser Verfahren beurteilen und einstufen.										
13. Inhalt:	Moderne Verfahren der einstigen "Geheimwissenschaft" Kryptographie werden eingeführt. Die Veranstaltung stellt Methoden zur Erzeugung elektronischer Unterschriften und zur Identifikation von Benutzern vor, die als notwendige Voraussetzungen für elektronische Wahlen oder anonymes elektronisches Bargeld gelten. Es werden neben klassischen, symmetrischen Verschlüsselungsverfahren aktuelle asymmetrische Verfahren behandelt. Eine wichtige Rolle spielen Protokolle, die aufbauend auf kryptographischen Verfahren die erwähnten Aufgaben lösen.										
14. Literatur:	• Bruce Schneier, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C, 1996 • Douglas Robert Stinson, Cryptography: Theory and Practice, 1995 • Friedrich Ludwig Bauer, Entzifferte Geheimnisse: Methoden und Maximen der Kryptologie, 1995 • Johannes Buchmann, Einführung in die Kryptographie, 1999										
15. Lehrveranstaltungen und -formen:	294601 Vorlesung mit Übungen Kryptographische Verfahren										
16. Abschätzung Arbeitsaufwand:	<table><tr><td>Präsenzzeit:</td><td>42 h</td></tr><tr><td>Selbststudiums- /</td><td>138 h</td></tr><tr><td>Nachbearbeitungszeit:</td><td></td></tr><tr><td>Summe:</td><td>180 h</td></tr></table>			Präsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Präsenzzeit:	42 h										
Selbststudiums- /	138 h										
Nachbearbeitungszeit:											
Summe:	180 h										
17. Prüfungsnummer/n und -name:	• 29461 Kryptographische Verfahren (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich oder mündlich										
18. Grundlage für ... :											
19. Medienform:											
20. Angeboten von:	Theoretische Informatik										

Modul: 29480 Loose Coupling and Message Based Applications

2. Modulkürzel:	052010003	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Understand the problem of application integration and the fundamental concept of loose coupling. The pros and cons of messaging are clear, and the architecture of Message Oriented Middleware is understood. Key patterns of using messaging to solve (enterprise) application integration problems are understood.		
13. Inhalt:	Messaging is a cornerstone of the integration of heterogeneous applications inside and among enterprises. Applications that need to share data synchronously or asynchronously with each other can be made to interoperate by means of the feature-rich Message-Oriented Middleware (MOM) that has grown ubiquitous in enterprises. During this course we treat the approaches and challenges of application integration through messaging. At first, we will address concepts such as (a-)synchronous messaging and the different messaging styles, e.g. point-to-point and publish-subscribe, that are the foundation of message-based application integration. Later in the course we will take an in-depth look at the mechanics and architecture of MOM, in particular of the Java Messaging Service (JMS), which will also be used in examples and exercises. Throughout the course we will discuss and apply extensively Enterprise Application Integration (EAI) patterns. Especially, endpoint patterns, routing patterns, transformation patterns, messaging patterns, channel patterns, and management patterns will be presented; the composability of these patterns will be explained.		
14. Literatur:	G. Hohpe and B. Woolf: "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions." Addison-Wesley Professional, ISBN-13: 978-0321200686. October 2003. M. Hapner et al: "Java Messaging Service API Tutorial & Reference". Addison-Wesley 2001.		
15. Lehrveranstaltungen und -formen:	294801 Vorlesung mit Übungen Lose Kopplung & Message-basierte Integration		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		

17. Prüfungsnummer/n und -name:	29481 Loose Coupling and Message Based Applications (PL), schriftlich oder mündlich, 60 Min., Gewichtung: 1.0, schriftlich (60 min) oder mündlich (30 min)
18. Grundlage für ... :	
19. Medienform:	Lecture and accompanying exercises
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 29470 Machine Learning

2. Modulkürzel:	051200112	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	Marc Toussaint		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Solid knowledge in Linear Algebra, probability theory and optimization. Fluency in at least one programming language.		
12. Lernziele:	Students will acquire an in depth understanding of Machine Learning methods. The concepts and formalisms of Machine Learning are understood as generic approach to a variety of disciplines, including image processing, robotics, computational linguistics and software engineering. This course will enable students to formalize problems from such disciplines in terms of probabilistic models and the derive respective learning and inference algorithms.		
13. Inhalt:	Exploiting large-scale data is a central challenge of our time. Machine Learning is the core discipline to address this challenge, aiming to extract useful models and structure from data. Studying Machine Learning is motivated in multiple ways: 1) as the basis of commercial data mining (Google, Amazon, Picasa, etc), 2) a core methodological tool for data analysis in all sciences (vision, linguistics, software engineering, but also biology, physics, neuroscience, etc) and finally, 3) as a core foundation of autonomous intelligent systems (which is my personal motivation for research in Machine Learning). This lecture introduces to modern methods in Machine Learning, including discriminative as well as probabilistic generative models. A preliminary outline of topics is: • motivation and history • probabilistic modeling and inference • regression and classification methods (kernel methods, Gaussian Processes, Bayesian kernel logistic regression, relations) • discriminative learning (logistic regression, Conditional Random Fields) • feature selection • boosting and ensemble learning • representation learning and embedding (kernel PCA and derivatives, deep learning) • graphical models • inference in graphical models (MCMC, message passing, variational) • learning in graphical models • structure learning and model selection • relational learning		

Please also refer to the course web page: <http://ipvs.informatik.uni-stuttgart.de/mlr/marc/teaching/13-MachineLearning/>

14. Literatur:	[1] <i>The Elements of Statistical Learning: Data Mining, Inference, and Prediction</i> by Trevor Hastie, Robert Tibshirani and Jerome Friedman. Springer, Second Edition, 2009. full online version available: http://www-stat.stanford.edu/~tibs/ElemStatLearn/ (recommended: read introductory chapter) [2] <i>Pattern Recognition and Machine Learning</i> by Bishop, C. M.. Springer 2006. online: http://research.microsoft.com/en-us/um/people/cmbishop/prml/ (especially chapter 8, which is fully online)
15. Lehrveranstaltungen und -formen:	• 294701 Lecture Machine Learning • 294702 Exercise Machine Learning
16. Abschätzung Arbeitsaufwand:	Presence time: 42 hours Self study: 138 hours Sum: 180 hours
17. Prüfungsnummer/n und -name:	• 29471 Machine Learning (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Parallele und Verteilte Systeme

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:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Knowledge of at least one programming language and knowledge in the field of computer science or similar subjects. Kenntnisse in mindestens einer Programmiersprache und in mindestens einem Fach aus dem Bereich der Technischen Informatik oder ähnlichen Fächern.		
12. Lernziele:	Students are able to master the practical programming of microcontrollers and are familiar with classical architectures. Historical Overview Microcontroller architectures Applications of microcontrollers Instruction set classic microcontroller Assembly language programming of microcontrollers C programming for microcontrollers Studierende beherrschen die praktische Programmierung von Mikrocontrollern und kennen klassische Architekturen. • Historische Übersicht • Mikrocontroller-Architekturen • Einsatzgebiete von Mikrocontrollern • Befehlssatz klassischer Microcontroller • Assembler-Programmierung von Mikrocontrollern • C-Programmierung von Mikrocontrollern		
13. Inhalt:	Microcontrollers (also called μController, μC, MCU) are IC's that combine at least peripheral functions on a single chip. In many cases, the working and programming memory is also partially or completely on the same chip. A microcontroller is practically a one-chip computer system. The number of built-in microcontroller exceeds by far the number of microprocessors. A microcontroller is often part of an embedded system in devices of everyday life like washing machines, smart cards (money, telephone cards), consumer electronics (VCRs, disc players, radios, televisions, remote controls), office electronics, motor vehicles (ECU for ABS, airbag, engine,		

instrument cluster, ESP, etc.), mobile phones and even in clocks and watches. In addition they are found on virtually all computer peripherals including keyboards, mouse, printers, monitors, scanners etc.

Microcontrollers are adapted to performance and respective features of the application. Therefore they have significant advantages in cost and power consumption compared with normal computers.

Small microcontrollers are available in high numbers for less than 1\$.

Als Microcontroller (auch μ Controller, μ C, MCU) werden ICs bezeichnet, die mit dem Prozessor mindestens Peripheriefunktionen auf einem Chip vereinen. In vielen Fällen befindet sich der Arbeits- und Programmspeicher 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 Mikroprozessoren.

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 1€, - 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 Gesamt: 180 Stunden
17. Prüfungsnummer/n und -name:	29641 Mikrocontroller (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündlichen Prüfung von 30 Min.
18. Grundlage für ... :	
19. Medienform:	

20. Angeboten von:

Modul: 29720 Mobile Computing

2. Modulkürzel:	051200166	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	• Kurt Rothermel • Frank Dürr		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Rechnernetze		
12. Lernziele:	The knowledge that has been acquired in the course "Computer Networks I" regarding concepts, protocols, and technologies of computer networks , will be extended to mobile devices and wireless communication systems and procedures. The objective of this lecture is to understand problems that might occur in the usage of mobile devices as well as to obtain knowledge to develop solutions for these problems and to communicate with experts. The Participants will learn about advantages and the disadvantages of specific wireless communication technologies for mobile devices and will be able to use appropriate protocols for the applications or modify them as needed. The exercises are used to provide practical experience in programming, analysis, performance evaluation of mobile and wireless communication systems as well as the expertise in the usage of appropriate tools.		
13. Inhalt:	1. Fundamentals of wireless data transmission 2. Media access for wireless networks 3. Location Management 4. Wireless Wide Area Networks 5. Wireless networks (local/personal) 6. Ad-hoc Networks: Exchange, Location administration 7. Mobility in IP-networks 8. Transport layer protocols for mobile systems 9. Location of services 10. Mobile data access 11. Introduction 12. Wireless data transmission 13. Location Management 14. Wireless 15. Telephone communication systems : GSM, GPRS,UMTS 16. Wireless networks (local/personal): 802.11, Bluetooth 17. Ad-hoc Networks: Routing, Location Management 18. Internetworking: Mobile IP, Cellular IP 19. Transport layers for mobile systems 20. Location of services : Problem, JINI, UpnP 21. Mobile data access: Broadcast Scheduling, Hoarding		

14. Literatur:	Charles E. Perkins: Mobile IP: Design Principles and Practices. 1997 James D. Solomon: Mobile IP: The Internet Unplugged. 1998 Jochen Schiller: Mobile Communications. 2000 Jörg Roth: Mobile Computing: Grundlagen, Technik und Konzepte. 2002 Kian-Lee Tan, Beng-Chin Ooi: Data Dissemination in Wireless Computing Envi-ronments. 2000 Tomasz Imielinski, Henry F. Korth (ed.): Mobile Computing. 1996
15. Lehrveranstaltungen und -formen:	297201 Vorlesung mit Übung Mobile Computing
16. Abschätzung Arbeitsaufwand:	Mobile Computing Vorlesung - Präsenzzeit: 21 Stunden - Selbststudium: 69 Stunden Mobile Computing Übungen - Präsenzzeit: 21 Stunden - Selbststudium: 69 Stunden
17. Prüfungsnummer/n und -name:	• 29721 Mobile Computing (PL), schriftlich oder mündlich, Gewichtung: 1.0, Prüfungsdauer: 90 min schriftlich oder 30 min mündlich Exam duration: 90 min written exam or 30 min oral exam • V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	Folien, Tafel
20. Angeboten von:	Verteilte Systeme

Modul: 10120 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:	Dirk Pflüger		
9. Dozenten:	• Dirk Pflüger • Stefan Zimmer • Miriam Mehl		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	• Modul 10190 Mathematik für Informatiker und Softwaretechniker • Modul 10240 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 diskrete Modelle sowie deren Behandlung, aber auch kontinuierliche Modelle werden ergänzend gestreift. Ob diskrete Ereignissimulation, spieltheoretische Ansätze, Zelluläre Automaten, Räuber-Beute Modelle oder Fuzzy-Mengen: die verschiedenen Modellierungsansätze sind so vielfältig wie die Problemstellungen, auf die sie angewendet werden. Verkehrssimulation, Populationswachstum, Wahlen oder Regelung sind nur einige der Anwendungsbereiche aus den Natur- und Ingenieurwissenschaften.		
14. Literatur:	• Modellbildung und Simulation - Eine anwendungsorientierte Einführung; Bungartz, H.-J., Zimmer, S., Buchholz, M., Pflüger, D., Springer Verlag, eXamen.press, 2013, ISBN 978-3-642-38656-6		
15. Lehrveranstaltungen und -formen:	• 101201 Vorlesung Modellbildung und Simulation • 101202 Übung Modellbildung und Simulation		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	10121 Modellbildung und Simulation (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Simulation großer Systeme

Modul: 29730 Modelling, Simulation, and Specification

2. Modulkürzel:	051711020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Master-level understanding of fundamental models of computation and their simulation, ability to apply them to embedded systems specification.		
13. Inhalt:	Given the complexity and implementation cost of contemporary electronic systems, it is essential to specify their intended functionality before elaborating the implementation. This course focuses on the model-based and executable specification of embedded systems and covers the following topics: • Hierarchical concurrent state machine models; • Kahn process networks, synchronous data flow networks; • Models of computation; • Tagged signal model; • Specification of timing, concurrency, and non-functional aspects; • Event-driven simulation; • Statically scheduled simulation; • Parallel simulation techniques		
14. Literatur:	• Lecture Notes "Modelling, Simulation, and Specification". • Jantsch: Modeling Embedded Systems and SoCs Concurrency and Time in Models of Computation. Morgan Kaufman Publishers, 2004. • Black, D.; Donovan, D.: SystemC from the Ground Up. Kluwer Academic Publishers, 2004.		
15. Lehrveranstaltungen und -formen:	• 297301 Vorlesung Modelling, Simulation, and Specification • 297302 Übung Modelling, Simulation, and Specification		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Summe: 180 Stunden		

17. Prüfungsnummer/n und -name:
- 29731 Modelling, Simulation, and Specification (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0
 - V Vorleistung (USL-V), schriftlich oder mündlich, Als Zulassungsvoraussetzung zur Klausur ist die folgende Vorleistung zu erbringen: Teilnahme an den Übungen, Präsentation der Lösung wenigstens einer Aufgabe.
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Eingebettete Systeme (Embedded Systems Engineering)

Modul: 55650 Multimodal Interaction for Ubiquitous Computers

2. Modulkürzel:	051900033	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Albrecht Schmidt		
9. Dozenten:	• Albrecht Schmidt • Niels Henze		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Basics of human computer interaction		
12. Lernziele:	Broad understanding for methods and concepts of multimodal interactions of personal computers, in particular for mobile systems, vehicles, tedious devices and environments.		
13. Inhalt:	• Interaction with mobile phones • User interfaces for vehicles • Interaction with intelligent environments • Interactive interfaces and gestures • Tangible user interfaces • Speech input and output • Camera-based interaction • Physiological sensors as interfaces between human and computer • Activities, context and emotions as input • Methods and techniques for designing user interfaces • Approaches for evaluating user interfaces		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	• 556501 Lecture Multimodal Interaction for Ubiquitous Computers • 556502 Exercise Multimodal Interaction for Ubiquitous Computers		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	55651 Multimodal Interaction for Ubiquitous Computers (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme		

Modul: 42460 Numerische Simulation

2. Modulkürzel:	051240060	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	• Dirk Pflüger • Stefan Zimmer • Miriam Mehl		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	• Modul 10190 Mathematik für Informatiker und Softwaretechniker und • Modul 10240 Numerische und Stochastische Grundlagen der Informatik bzw. • Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker • Modul 42410 Grundlagen des wissenschaftlichen Rechnens		
12. Lernziele:	Fähigkeit zur Implementierung numerischer Methoden und Entwicklung und Umsetzung geeigneter Datenstrukturen.		
13. Inhalt:	Strukturmechanik, Strömungsmechanik, Finite Elemente, Finite Differenzen sowie praktische Aspekte der effizienten und parallelen Umsetzung auf Rechnern.		
14. Literatur:	• Griebel, Dornseifer, Neunhoeffter: Numerical simulation in fluid dynamics : a practical introduction; SIAM, 1998 / Numerische Simulation in der Strömungsmechanik; Vieweg 1995 • Griebel, Knapek, Zumbusch, Caglar: Numerische Simulation in der Moleküldynamik : Numerik, Algorithmen, Parallelisierung, Anwendungen; Springer 2004 • Braess: Finite Elemente : Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie; Springer, 2007		
15. Lehrveranstaltungen und -formen:	• 424601 Vorlesung Numerische Simulation • 424602 Übung Numerische Simulation		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	42461 Numerische Simulation (LBP), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Simulation großer Systeme		

Modul: 56790 Parallele Numerik

2. Modulkürzel:	051240080	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Miriam Mehl		
9. Dozenten:	• Miriam Mehl • Dirk Pflüger • Stefan Zimmer		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	• Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker oder • Modul 10240 Numerische und Stochastische Grundlagen		
12. Lernziele:	Die Studenten kennen die wesentlichen parallelisierbaren Algorithmen für zentrale numerische Problemstellungen. Sie erkennen Parallelisierungshindernisse in bekannten und neuen numerischen Algorithmen, können die zu erwartende Skalierbarkeit abschätzen und sind in der Lage, Algorithmen so zu modifizieren, dass die parallele Effizienz erhöht wird ohne wichtige numerische Eigenschaften wie Stabilität und Komplexität zu verlieren.		
13. Inhalt:	• parallele Matrix- und Vektoroperationen • parallele Fouriertransformation • parallele QR Zerlegung und Least Squares Probleme • parallele iterative Gleichungssystemlöser • parallele Eigenwert- und Eigenvektorberechnung • parallele Zeitschrittverfahren • parallele Algorithmen für Teilchenwechselwirkungen		
14. Literatur:	• Introduction to High Performance Scientific Computing (Eijkhout, Chow, van de Geijn) (download at http://www.lulu.com/shop/victor-eijkhout/introduction-to-high-performance-scientific-computing/paperback/product-21431780.html;jsessionid=CF30CC0B65B0F349BFBD206D406F8) • Numerical Linear Algebra for High-Performance Computers (Dongarra, Duff, Sorensen, van der Vorst) • Parallel Algorithms for Matrix Computations (Gallivan, Heath, Ng, Ortega,...) • A User's Guide to MPI (Pacheco) • Iterative Methods for Sparse Linear Systems (Saad) • Loesung linearer Gleichungssysteme auf Parallelrechnern (Frommer) • M. Griebel, S. Knapek, G. Zumbusch, and A. Caglar. Numerische Simulation in der Molekulardynamik. Springer, 2004. • D. Frenkel and B. Smith. Understanding Molecular Simulation from Algorithms to Applications. Academic Press (2nd ed.), 2002.		

15. Lehrveranstaltungen und -formen:	• 567901 Vorlesung Parallele Numerik • 567902 Übung Parallele Numerik	
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h
	Selbststudiums- /	138 h
	Nachbearbeitungszeit:	
	Summe:	180 h
17. Prüfungsnummer/n und -name:	56791 Parallele Numerik (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0	
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:	Simulation großer Systeme	

Modul: 29650 Parallele Programmierung

2. Modulkürzel:	051230130	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Kenntnisse in mindestens einer Programmiersprache.Kenntnisse in mindestens einem Fach der Technischen Informatik odereinem ähnlichen Fach.		
12. Lernziele:	Studierende beherrschen die Programmierung von Multi-Core Prozessoren und paralleler Rechner bzw. Computing-Systemen.		
13. Inhalt:	• Grundlegende Parallelisierungsansätze: Parallelisierung durch Datenzerlegung, parallele lineare Algebra, etc. • Message Passing Interface • Open MP • C-Programmierung für FPGAs • Graphische Programmierung • GPU-Programmierung		
14. Literatur:	• Thomas Rauber und Gundula Rünger, Multicore: Parallele Programmierung (Informatik Im Fokus), 2007 • More literature is named in the lecture		
15. Lehrveranstaltungen und -formen:	296501 Vorlesung mit Übung Parallele Programmierung		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29651 Parallele Programmierung (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 10250 Parallele Systeme

2. Modulkürzel:	051200065	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Erfahrungen 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 klassischen Mikroprozessor zur Multi-Core CPU Programmierung paralleler Rechnersysteme • Systolische Arrays, massiv parallele Systeme • Parallele Systeme aus verschiedenen Anwendungsdomänen: ausgewählte Fallbeispiele		
14. Literatur:	Wird in der Lehrveranstaltung bekannt gegeben.		
15. Lehrveranstaltungen und -formen:	• 102501 Vorlesung Parallele Systeme • 102502 Übung Parallele Systeme		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	10251 Parallele Systeme (LBP), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 48550 Practical Course Information Systems

2. Modulkürzel:	051200135	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Bernhard Mitschang		
9. Dozenten:	• Bernhard Mitschang • Holger Schwarz		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 3. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 3. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 3. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Basic knowledge on database systems, information systems and programming languages		
12. Lernziele:	Students get hands-on experience with state-of-the-art information systems. Students learn how to use these systems to address typical tasks in information processing. Based on this practical experience, they will also be able to assess available technologies and systems for various application areas.		
13. Inhalt:	The focus of this course is on the design and implementation of database-oriented applications. This includes core database technology as well as middleware and web technology.		
14. Literatur:	Will be announced at the beginning of the course		
15. Lehrveranstaltungen und -formen:	485501 Informationssystem-Fachpraktikum		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 60 Stunden Selbststudium: 120 Stunden		
17. Prüfungsnummer/n und -name:	48551 Practical Course Information Systems (LBP), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Datenbanken und Informationssysteme		

Modul: 48570 Practical Course Visual Computing

2. Modulkürzel:	051900111	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	Thomas Ertl		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Basics of Computer Graphics		
12. Lernziele:	During this practical course, students will learn about approaches to rendering and visual computing technologies and will know how to implement these. They will learn about polygon based approach as well as volume rendering approaches. The students will learn, how to proceed a small project on their own (independently).		
13. Inhalt:	• OpenGL • Qt-Framework • Raytracing • Volume Rendering • Independent Project		
14. Literatur:	• OpenGL Programming Guide - Third Edition (OpenGL 1.2) , Masonn Woo, Jackie Neider, Tom Davis, Dave Shreiner, Addison Wesley, 1999 • Programming with Qt - First Edition, Matthias Kalle Dalheimer, O'Reilly, 1999 • An Introduction to Ray Tracing, Andrew S. Glassner, Academic Press, 1989 • Computer Graphics - Principle and Practice - Second Edition, Foley, van Dam, Feiner, Huges, Addison Wesley, 1990		
15. Lehrveranstaltungen und -formen:	485701 Lab Practical Course Visual Computing		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	
17. Prüfungsnummer/n und -name:	48571 Practical Course Visual Computing (LBP), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme		

Modul: 29660 Programmanalysen und Compilerbau

2. Modulkürzel:	051510311	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Kenntnisse, die in etwa den Inhalten des Moduls 10150 - Grundlagen des Compilerbaus und der Programmiersprachen - des Bachelor-Studiums entsprechen, sind dringend empfohlen.		
12. Lernziele:	Die Studierenden haben grundlegende Kenntnisse über die typischen in Compilern und anderen statischen Programmanalysen verwandten Verfahren erworben, sowohl in Bezug auf Basisanalysen (Kontroll- und Datenflussanalysen) als auch auf weitergehende, zielgerichteten Analysen wie Zeigeranalysen, Abhängigkeitsanalysen oder Slicing. Speziell lernen sie eine Reihe von Codeoptimierungen im Compiler kennen, aber auch diverse Globalanalysen, wie sie in Werkzeugen zur Fehlersuche, zum Reengineering oder zu Architekturanalysen nötig sind.		
13. Inhalt:	• Attributgrammatiken (Wiederholung) • Programmanalysen und -Optimierung (Schwerpunkt) • klassische Optimierungen • Lokale und globale Kontrollflussanalyse • Lokale und globale Datenflussanalysen • Dominatoren, Dominatorgrenzen, Kontrollstrukturanalysen • Zeigeranalysen • Seiteneffekt-Analyse • Datenabhängigkeiten, Konfliktanalysen und Registervergabe • SSA-Form und ihre Berechnung • Code-Erzeugung • Implementierung von OOP • Das Laufzeitsystem • Separate Übersetzung • Slicing • Mustersuchen und Klonerkennung • Begriffsanalyse und ihre Anwendungen Orthogonal zu den jeweiligen Analyseverfahren werden die Verwendungen in Codeoptimierung und in Programmanalysen anderer Werkzeuge des Software Engineering aufgezeigt.		
14. Literatur:	• A. V. Aho, M.S. Lam, R. Sethi, J. D. Ullman: Compilers - Principles, Techniques, and Tools, Addison Wesley Verlag (2007) • Morgan, Robert, Building an Optimizing Compiler, 1998		

- Muchnick, Steven S., Advanced Compiler Design and Implementation, 1997
- Wilhelm, Maurer: Übersetzerbau, 2. Auflage, Springer Verlag (1997)
- Uwe Kastens: Übersetzerbau, Oldenbourg Verlag (1990)

15. Lehrveranstaltungen und -formen:	296601 Vorlesung mit Übung Programmanalysen und Compilerbau								
16. Abschätzung Arbeitsaufwand:	<table> <tr> <td>Präsenzzeit:</td><td>42 h</td></tr> <tr> <td>Selbststudiums- /</td><td>138 h</td></tr> <tr> <td>Nachbearbeitungszeit:</td><td></td></tr> <tr> <td>Summe:</td><td>180 h</td></tr> </table>	Präsenzzeit:	42 h	Selbststudiums- /	138 h	Nachbearbeitungszeit:		Summe:	180 h
Präsenzzeit:	42 h								
Selbststudiums- /	138 h								
Nachbearbeitungszeit:									
Summe:	180 h								
17. Prüfungsnummer/n und -name:	29661 Programmanalysen und Compilerbau (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0								
18. Grundlage für ... :									
19. Medienform:									
20. Angeboten von:	Institut für Softwaretechnologie								

Modul: 51740 Quantencomputing

2. Modulkürzel:	050420210	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	Ulrich Hertrampf		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Grundkenntnisse in Mathematik und Theoretischer Informatik, wie sie in „Mathematik für Informatiker“ und „Theoretische Grundlagen der Informatik“ vermittelt werden.		
12. Lernziele:	Die Studierenden lernen die Grundprinzipien kennen, nach denen der im wesentlichen noch immer hypothetische Quantencomputer arbeiten würde. Sie kennen die grundlegenden Quantenalgorithmen von Deutsch, Jozsa, Simon und anderen, den Zusammenhang mit unitären Matrizen, wichtige Sätze wie das No-Cloning-Theorem. Sie haben den Quanten-Suchalgorithmus von Grover und den Primfaktorzerlegungs-Algorithmus von Shor verstanden.		
13. Inhalt:	Die Vorlesung behandelt zunächst die grundlegenden Techniken des Quanten-Computings: Qubits und Quantenregister, Messungen, Hadamard-Transformation, Quantenschaltkreise. Im zweiten Teil werden wichtige Algorithmen vorgestellt: Grovers Suchalgorithmus, Shors Faktorisierungsalgorithmus. Abschnitte über Teleportation und Quantenkryptographie runden das Thema ab.		
14. Literatur:	Matthias Homeister, „Quantum Computing verstehen“, 2. Auflage, Friedr. Vieweg & Sohn, 2008 Jozef Gruska, „Quantum computing“, McGraw-Hill, 1999.		
15. Lehrveranstaltungen und -formen:	517401 Vorlesung mit Übungen Quantencomputing		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 h Selbststudiums- / 138 h Nachbearbeitungszeit: Summe: 180 h		
17. Prüfungsnummer/n und -name:	51741 Quantencomputing (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Formale Methoden der Informatik		

Modul: 29670 Rapid Prototyping

2. Modulkürzel:	051230135	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Erfahrungen in mindestens einer Programmiersprache.		
12. Lernziele:	Die Studierenden beherrschen die schnelle Realisierung von Computing-Systemenausgehend von einer Algorithmen-Implementierung unter Verwendung eines Computer-Algebrasystems.		
13. Inhalt:	Text		
14. Literatur:	• James O. Hamblen und Michael D. Furman, Rapid Prototyping of Digital Systems: A Tutorial Approach, 2001 • More literature is named in the lecture		
15. Lehrveranstaltungen und -formen:	296701 Vorlesung mit Übung Rapid Prototyping		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29671 Rapid Prototyping (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 29680 Real-Time Programming

2. Modulkürzel:	051510301	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	• Significant programming experience (not necessarily in real-time application) is highly advisable. • Knowledge of Ada, C/C++ and Unix is helpful, but not required.		
12. Lernziele:	Students understand the standard terminology of deadline-driven, safety-critical real-time systems. They understand the issues that differentiate such systems from general software systems, and they know about available solutions, if any.		
13. Inhalt:	1) General requirements and terminology of real-time systems 2) Deterministic execution: avoiding language-, implementation- and hardware-induced non-determinisms; coping with limited resources; storage estimation and management; execution time estimation 3) Fault tolerance: Faults and failure modes, N-version programming, voting, forward and backward recovery 4) Simple scheduling regimes: cyclic executives, deadline guarantees 5) Parallelism and priority scheduling regimes: processes, threads, tasks; run-time kernels; task management; interrupt handling 6) Synchronization and communication: semaphores, critical regions, monitors, protected objects, rendezvous, messaging 7) Control of shared resources 8) Distributed Systems: basic concepts; major issues		
14. Literatur:	• Alan Burns and Andy Wellings: Real-Time Systems and Programming Languages, Addison Wesley, 1997 ... or later editions of the Burns/Wellings-Book, e.g., 4.ed. 2009 • Language reference manuals (C++, Java, Ada) are useful at times.		
15. Lehrveranstaltungen und -formen:	296801 Vorlesung mit Übung Real-Time Programming		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 h Selbststudiums- / 138 h Nachbearbeitungszeit: Summe: 180 h		
17. Prüfungsnummer/n und -name:	29681 Real-Time Programming (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von: Institut für Softwaretechnologie

Modul: 29690 Real-Time Video Processing I

2. Modulkürzel:	051230140	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	This course requires knowledge and experience in (at least) one programming language as well as knowledge of the subject "Technische Informatik or a similar course"		
12. Lernziele:	The Students will gain knowledge in the implementation of algorithms, architectures and exemplary processors for real-time video processing		
13. Inhalt:	• Introduction: analog/digital Television • Cameras, Image sensors and their characteristics • Image Filtering, Bayer Filter • Motion Analysis • video compression • video communication • video processing • Parallel architecture, video processors and Implementation of hardware components for real-time video processing algorithms		
14. Literatur:	• Roger Clarke und R. J. Clarke von Academic Press Inc, Digital Compression of Still Images and Video (Signal Processing and Its Applications), 1995 • More literature is named in the lecture		
15. Lehrveranstaltungen und -formen:	296901 Vorlesung mit Übung Real-Time Video Processing I		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29691 Real-Time Video Processing I (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 29700 Real-Time Video Processing II

2. Modulkürzel:	051230142	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Voraussetzung für Real-Time Video Processing II sind Kenntnisse von Real-Time Video Processing I. Alternativ sind Kenntnisse aus einem Fach der Technischen Informatik oder einem ähnlichen Fach oder Kenntnisse im Bereich der Datenkompression oder der Bildverarbeitung oder der Signalverarbeitung Voraussetzung.		
12. Lernziele:	Die Studierenden können praktisch Real-Time Video Processing Systeme aufbauen.		
13. Inhalt:	• Vorstellung der Fallstudie eines Video Processing Systems • Auswahl der Algorithmen des Video Processing Systems • Implementierung und Verifikation der Algorithmen • Architektur-Entwicklung des Video Processing Systems • Performance-Analyse der Achitektur • Implementierung und System-Verifikation		
14. Literatur:	• Roger Clarke und R. J. Clarke von Academic Press Inc, Digital Compression of Still Images and Video (Signal Processing and Its Applications), 1995 • More literature is named in the lecture		
15. Lehrveranstaltungen und -formen:	297001 Vorlesung mit Übung Real-Time Video Processing II		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29701 Real-Time Video Processing II (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 45740 Rechnernetze II

2. Modulkürzel:	051200168	5. Moduldauer:	2 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	• Kurt Rothermel • Frank Dürr		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Rechnernetze		
12. Lernziele:	Das in der Veranstaltung Rechnernetze I erworbene Verständnis von Konzepten, Protokollen und Technologien von Rechnernetzen wird weiter vertieft. Der Teilnehmer kennt die Funktionsweise der wichtigsten Dienste der Anwendungsschicht des Schichtenmodells. Er kennt die Konzepte zur Realisierung von Netzen auf Anwendungsebene. Er ist im Stande, diese Dienste und Konzepte bei der Konzeption eigener Anwendungen zu nutzen und ebenso eigene Dienste und Systeme zu entwickeln, um konkrete Probleme zu lösen.		
13. Inhalt:	Vorlesung Höhere Kommunikationskonzepte und -Protokolle: 1. Einführung 2. Socket-Schnittstelle 3. Präsentation und Kompression 4. Realzeitkommunikation 5. Elektronische Bezahlssysteme 6. Multicast auf Anwendungsschicht 7. Inhaltsbezogene Netze 8. Geographische Kommunikation Vorlesung Peer-to-Peer-Systeme: 1. Definition und Anwendungen von Peer-to-Peer-Systemen 2. Theoretische Netzmodelle 3. Unstrukturierte Peer-to-Peer-Systeme 4. Strukturierte Peer-to-Peer-Systeme 5. Komplexe Suchanfragen 6. Sicherheitsmechanismen für Peer-to-Peer-Systeme 7. Software-Rahmenwerke für Peer-to-Peer-Systeme		
14. Literatur:	• J.F. Kurose, K.W. Ross, Computer Networking. 4th Edition, 2007 • L.L. Peterson, B.S. Davie, Computer Networks: A Systems Approach. 4th Edition, 2007		

- Peter Mahlmann, Christian Schindelbauer, P2P Netzwerke: Algorithmen und Methoden., 2007
- Ralf Steinmetz, Klaus Wehrle, Peer-to-Peer Systems and Applications, 2005A.S. Tanenbaum, Computer Networks, 4th Edition, 2003

15. Lehrveranstaltungen und -formen:	• 457401 Vorlesung Höhere Kommunikationskonzepte und -protokolle • 457402 Vorlesung Peer-to-Peer-Systeme
--------------------------------------	--

16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden
---------------------------------	---

Gesamt: 180 Stunden

17. Prüfungsnummer/n und -name:	45741 Rechnernetze II (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Prüfungsdauer: 90 min schriftlich oder 30 min mündlich Exam duration: 90 min written exam or 30 min oral exam
---------------------------------	---

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:	Verteilte Systeme
--------------------	-------------------

Modul: 48620 Scientific Visualization

2. Modulkürzel:	051900777	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	• Thomas Ertl • Daniel Weiskopf • Steffen Frey		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule -->Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Basic concepts of Human Computer Interaction Basic concepts of Computer Graphics		
12. Lernziele:	Student gains expertise about fundamental concepts and techniques of scientific visualization. This includes algorithms and mathematical background, data structures and implementation aspects as well as practical experience with widely available visualization tools.		
13. Inhalt:	Visualization discusses all aspects of visual representations of data gained from experiments, simulations, medical scanning machines, data bases an the like. The aim of visualization is to gain further insights into the data or the generate "simple" representations of complex phenomena or issues. For that, known techniques from the research area of interactive computer graphics as well as novel techniques are applied. The following topics will be discussed: • Introduction, history, visualization pipeline • Data aquisition and representation (sampling, reconstruction, grids, data structures) • PerceptionBasic concepts of visual mappings • Visualization of scalar fields (extraction of iso-surfaces, volume rendering) • Visualization of vector fields (particle tracking, texture-based methods, topology) • Tensor fields, multivariate data • Highdimensional data and information visualization		
14. Literatur:	• C. D. Hansen, C. R. Johnson, The Visualization Handbook, 2005 • C. Ware, Information Visualization: Perception for Design, 2004		
15. Lehrveranstaltungen und -formen:	• 486201 Lecture Scientific Visualization • 486202 Exercise Scientific Visualization		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiums- /	138 h	
	Nachbearbeitungszeit:		
	Summe:	180 h	

17. Prüfungsnummer/n und -name:
- 48621 Scientific Visualization (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0
 - V Vorleistung (USL-V), schriftlich oder mündlich
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

Modul: 29510 Service Computing

2. Modulkürzel:	052010004	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	A diversity of technologies enable nowadays computer-based interactions on the Web and on the Internet. The aim of this course is to make the students familiar with some of the most pervasive technologies that come together to form the Web and the Internet as we know it, and that enable to build large-scale application systems.		
13. Inhalt:	At first, we will cover the Web-centric technologies that enable the interaction of humans with Web content, e.g. HTTP, SMTP, AJAX, CSS and MIME . On the server-side part of technology, we will treat several Java EE technologies such as portlets, servlets, and JSP. The second part of the course will cover a set of technologies that are prominent in the landscape of Service-Oriented Architecture (SOA). In a nutshell, SOA is a paradigm that advocates the creation of complex, value added applications by reusing and composing independent and loosely coupled (software) services. We will dissect prominent SOA concepts like service discovery, addressing, policies, Service Bus, coordination protocols and service compositions. The architectural concepts will be complemented with an outlook of the technologies that embody them in the landscape of enterprise computing. In particular, we will cover several XML-centric technologies that sit at the core of Web services, e.g. XSD, SOAP, WSDL and Policy. In addition to the SOAP-based approach to Web services, we will also explore their REST aspect. Building on this portfolio of technologies, we will discuss the relationships between Web service technologies and “hot” items on the enterprise computing agenda such as autonomic/organic computing and cloud computing.		
14. Literatur:	S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson: “Web Services Platform Architecture”, Prentice Hall 2005 G. Alonso, F. Casati, H. Kuno, V. Machiraju: “Web Services”, Springer 2004 E. Wilde: “World Wide Web”, Springer 1999		

M.P. Papazoglou: "Web Services: Principles & Technology", Pearson Education Limited 2008

N.M. Josuttis: "SOA in Practice: The Art of Distributed System Design", O'Reilly 2007

Th. Erl: "SOA: Entwurfsprinzipien für serviceorientierte Architektur", Addison-Wesley 2008

D.A. Chappell: "Enterprise Service Bus", O'Reilley 2004

15. Lehrveranstaltungen und -formen:	295101 Vorlesung mit Übungen Service Computing
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	29511 Service Computing (PL), schriftlich oder mündlich, 60 Min., Gewichtung: 1.0, schriftlich (60 min) oder mündlich (30 min)
18. Grundlage für ... :	29520 Ausgewählte Themen des Service Computing
19. Medienform:	Lecture and accompanying exercises
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 46660 Service Management and Cloud Computing, and Evaluation

2. Modulkürzel:	052000111	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Kristof Klöckner		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Service Computing, Business Process Management		
12. Lernziele:	The students will learn the basics of systems management and cloud computing.		
13. Inhalt:	Cloud Computing is an emerging paradigm for consumption and delivery of IT based services, based on concepts derived from consumer internet services, like self-service, apparently unlimited or elastic resources and flexible sourcing options. In this course we will discuss the technical foundations of cloud computing, as well as the business models associated with it. We will start by looking at virtualization and service management as the technical underpinnings. We will then look at infrastructure services and platform services, with a particular focus on emerging programming models for the cloud. We will discuss the trade-offs made between consistency and availability as well as extensions to traditional programming models. We also look at the life-cycle of applications in the cloud. Finally, we will look some of the challenges of Software as a Service, like multi-tenancy. Throughout the course, we will look both at existing products and services as well as the theoretical underpinnings. The course will be held as a combination of lectures and participant discussion.		
14. Literatur:	To be announced in the lecture.		
15. Lehrveranstaltungen und -formen:	• 466601 Vorlesung Service Management and Cloud Computing, and Evaluation • 466602 Exercise Service Management and Cloud Computing, and Evaluation		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	• 46661 Service Management and Cloud Computing, and Evaluation (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), mündliche Prüfung, 30 Min.		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von:

Institut für Architektur von Anwendungssystemen

Modul: 60140 Sprachbau mit Language Workbenches

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Markus Völter		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 6. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Compilerbau Objektorientierte Programmierung		
12. Lernziele:	<i>Die Studenten verstehen, warum und wie Modelle für den praktizierenden Softwareentwickler nützlich sind: Kommunikation, Analyse, Synthese. Sie verstehen warum es in vielen Fällen sinnvoll ist, eigene (domänenspezifische) Sprachen zu bauen. Sie verstehen die Mechanismen um Sprachen zu bauen, insbesondere die Fähigkeiten moderner Language Workbenches. Die Studenten sind in der Lage mit JetBrains MPS Sprachen zu bauen.</i>		
13. Inhalt:	<i>Modellierung, Grammatiken, Projizierende Editoren, Typsysteme, Codegenerierung, Interpreter. Grundlagen des Sprachdesigns: Ausdrucksfähigkeit vs. Komplexität, Vollständigkeit, Modularisierung, verschiedene Notationen. Wichtige Sprachparadigmas, die man in DSLs wiederverwenden kann: imperativ, funktional, zustandsbasiert. Grundlagen der Arbeit mit MPS.</i> <i>Die Veranstaltung wird als Blockveranstaltung als Workshop ausgeführt, viele praktische Anteile. Die Klausur findet direkt am Ende der Blockveranstaltung statt.</i> <i>Der Zeitraum ist 22. Bis 26. August.</i> <i>Ort ist bei der itemis AG, Industriestrasse 6, Vaihingen (direkt neben dem Bhf)</i>		
14. Literatur:	Buch http://dslbook.org/ + ggfs. diverse wissenschaftliche Papiere		
15. Lehrveranstaltungen und -formen:	• 601401 Vorlesung Sprachbau • 601402 Übung Sprachbau		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit in Stunden: 56 Selbststudiumszeit in Stunden: 124		
17. Prüfungsnummer/n und -name:	60141 Sprachbau mit Language Workbenches (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		

18. Grundlage für ... :

19. Medienform: *Powerpoint, Tafel, Demos, Diskussionen, Selbstarbeit der Studenten*

20. Angeboten von: Software-Engineering

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:	Martin Fuchs		
9. Dozenten:			
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Ergänzungsmodule --> Wahlmodule aus Master Informatik → B.Sc. Informatik, PO 2012, 5. Semester → Vorgezogene Master-Module		
11. Empfohlene Voraussetzungen:	Modul 051900002 Computergraphik		
12. Lernziele:	The students know theoretical foundations for visual computing and acquired practical expertise in its core techniques. They are able to acquire scenes with digital cameras, can model their behavior and create content for non-2D displays and camera-projector systems.		
13. Inhalt:	The class is concerned with the digital processing of visual information by means of computer vision, computer graphics and image processing. It covers the following three interlocking topic complexes: Image processing: • mathematical basics of image representations • noise models and noise suppression (including morphological, bilateral, and non-local filters) • selected topics from discrete image processing on image regions (e.g. photo montage with graph cuts, texture synthesis and space-time video completion) Measuring / displaying light: • selected topics from simple optics (esp. thin lenses and their interactions with light) • geometric camera models and calibration, typical optical distortions and means to counter them • radiometric camera calibration and HDR imaging • measuring and displaying color • plenoptic imaging / integral photography techniques, light field rendering and light field displays • passive stereo Combined camera / illumination systems • camera - illumination systems and photometric stereo • active stereo and projector-camera systems • the light transport matrix, its measurement and applications Throughout, the class equally covers both acquisition (camera) and displays systems.		
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
	Gesamt: 180 Stunden
17. Prüfungsnummer/n und -name:	• 29501 Visual Computing (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündlichen 30 Min • V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

400 Schlüsselqualifikationen fachaffin

Zugeordnete Module: 10290 Projekt-INF
 10320 Seminar-INF 1
 42390 Seminar-INF 2

Modul: 10290 Projekt-INF

2. Modulkürzel:	051900095	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Niels Henze		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Schlüsselqualifikationen fachaffin B.Sc. Informatik, PO 2012, 5. Semester → Schlüsselqualifikationen fachaffin		
11. Empfohlene Voraussetzungen:	Basismodule der Informatik. Darüber hinaus variabel je nach Projektanforderung.		
12. Lernziele:	Die Studierenden sollen frühzeitig und beispielhaft an Informatik-Forschung herangeführt werden („undergraduate research“). Dazu soll in einem Team von mindestens 3 Studierenden in einem Zeitraum von höchstens 6 Monaten ein Projekt bearbeitet werden, das sich an aktuellen Forschungsfragestellungen der Abteilungen und Institute orientiert. Ein Beitrag zu laufenden Drittmittelprojekten ist möglich, ebenso eine Fortsetzung des Projekts in ausgewählten Bachelor-Arbeiten. Die Teilnehmer können ein forschungsorientiertes Projekt unter Anleitung planen, durchführen und die Ergebnisse dokumentieren und präsentieren. Teilnehmer verfügen insbesondere über die folgenden generischen Kompetenzen: 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 Projekte zu aktuellen Forschungsfragestellungen von den Prüfern des Fachbereichs Informatik angeboten. Die Themen haben einen überwiegenden Forschungscharakter, was sich aus dem Publikationspotential der erwarteten Ergebnisse ergibt. Die Projekte umfassen in der Regel: Einarbeitung und Literatursuche, Methodenentwicklung, Implementierung, Analyse, Dokumentation und Präsentation der Ergebnisse. Um dem Forschungscharakter des Projekts gerecht zu werden, soll das Ergebnis in einer wissenschaftlichen Publikation (max. 10 Seiten in Englisch) festgehalten werden, die einer einheitlichen Formatvorlage folgt. Einmal pro Semester sollen die bis zu einem Stichtag abgegebenen Projektpapiere auf einer internen Konferenz in einem Kurzbeitrag von den Studierenden präsentiert werden.		
14. Literatur:	wird in der Veranstaltung und im Web bekannt gegeben.		
15. Lehrveranstaltungen und -formen:	102901 Seminar Projekt		
16. Abschätzung Arbeitsaufwand:	180 Stunden pro Teammitglied		

17. Prüfungsnummer/n und -name: 10291 Projekt-INF (USL), Sonstiges, Gewichtung: 1.0, Scheinkriterien: Aktive Mitwirkung im Projektteam. Abgabe eines Projektberichts in Form einer wissenschaftlichen Publikation (max. 10 Seiten in Englisch) gemäß einer einheitlichen Formatvorlage. Teilnahme und Mitwirkung an der internen Semesterkonferenz.

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

Modul: 10320 Seminar-INF 1

2. Modulkürzel:	050420095	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes Semester
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester → Schlüsselqualifikationen fachaffin		
11. Empfohlene 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. Sie sind in der Lage, das von ihnen erarbeitete Thema auch schriftlich darzustellen.		
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:	Die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben.		
15. Lehrveranstaltungen und -formen:	103201 Seminar		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	21 h	
	Selbststudiums- /	69 h	
	Nachbearbeitungszeit:		
	Summe:	90 h	
17. Prüfungsnummer/n und -name:	10321 Seminar-INF 1 (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: Institut für Softwaretechnologie

Modul: 42390 Seminar-INF 2

2. Modulkürzel:	050420095	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes Semester
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Schlüsselqualifikationen fachaffin		
11. Empfohlene 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:	Die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben.		
15. Lehrveranstaltungen und -formen:	423901 Seminar		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	21 h	
	Selbststudiums- /	69 h	
	Nachbearbeitungszeit:		
	Summe:	90 h	
17. Prüfungsnummer/n und -name:	42391 Seminar-INF 2 (LBP), Sonstiges, Gewichtung: 1.0		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von: Institut für Softwaretechnologie

Modul: 81100 Bachelorarbeit Informatik

2. Modulkürzel:	050525002	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes Semester
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Funke		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 6. Semester B.Sc. Informatik, PO 2012, 6. Semester		
11. Empfohlene Voraussetzungen:	sämtliche Basis- und Kernmodule sowie mindestens 120 LP insgesamt im Bachelorstudium		
12. Lernziele:	Die Bachelorarbeit soll zeigen, dass innerhalb einer vorgegebenen Frist eine Aufgabenstellung aus dem Bereich Informatik selbständig nach wissenschaftlichen Methoden bearbeitet und die Ergebnisse sachgerecht dargestellt werden können.		
13. Inhalt:	wird vom Prüfer festgelegt		
14. Literatur:	wird von Prüfer bekanntgegeben		
15. Lehrveranstaltungen und -formen:			
16. Abschätzung Arbeitsaufwand:	Summe: 360 h		
17. Prüfungsnummer/n und -name:			
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			