Modulhandbuch

Studiengang Bachelor of Science Informatik

Prüfungsordnung: 079-2017

Sommersemester 2018
Stand: 09. April 2018

Universität Stuttgart
Keplerstr. 7
70174 Stuttgart

Kontaktpersonen:

Inhaltsverzeichnis

100 Basismodule 6

10190 Mathematik für Informatiker und Softwaretechniker 7
10280 Programmierung und Software-Entwicklung 9
12060 Datenstrukturen und Algorithmen 11
36530 Rechnerorganisation 1 13
78620 Theoretische Informatik I 14
78630 Theoretische Informatik II 15
78650 Technische Grundlagen der Informatik 16

200 Kernmodule 18

210 Pflichtmodule 19
78660 Programmierprojekt 20
78670 Numerische Grundlagen 21
78680 Statistische und stochastische Grundlagen 23
78730 Theoretische Informatik III 24
78740 Wissenschaftliche Methoden in der Informatik 25
220 Wahlpflichtmodule 26
2201 Praktische Informatik 27
10210 Mensch-Computer-Interaktion 28
10220 Modellierung 30
36100 Programmierparadigmen 32
40090 Systemkonzepte und -programmierung 34
2202 Wahlpflichtmodule allgemein 36
10060 Computergraphik 37
10170 Imaging Science 38
10210 Mensch-Computer-Interaktion 40
10220 Modellierung 42
14390 Programmentwicklung 44
16500 Software Engineering 45
17210 Einführung in die Softwaretechnik 46
36100 Programmierparadigmen 48
40090 Systemkonzepte und -programmierung 50
42410 Grundlagen des Wissenschaftlichen Rechnens 52
78640 Grundlagen der Informationssicherheit 53
78750 Rechnerorganisation 2 55

300 Ergänzungsmodule 56

310 Wahlmodule 57
10030 Architektur von Anwendungssystemen 58
10060 Computergraphik 60
10110 Grundlagen der Künstlichen Intelligenz 61
10170 Imaging Science 62
10180 Information Retrieval and Text Mining 64
10210 Mensch-Computer-Interaktion 65
10220 Modellierung 67
11620 Automatisierungstechnik I 69
14390 Programmentwicklung 71
16500 Software Engineering 72
17210 Einführung in die Softwaretechnik 73
36100 Programmierparadigmen 75
39040 Rechnernetze 77
40090 Systemkonzepte und -programmierung 79
42410 Grundlagen des Wissenschaftlichen Rechnens 81

78640 Grundlagen der Informationssicherheit 82
78750 Rechnerorganisation 2 84
320 Anwendungsfächer 85
3201 Anwendungsfach Kraftfahrzeugmechatronik 86
13590 Kraftfahrzeuge I + II 87
14130 Kraftfahrzeugmechatronik I + II 88
38370 Grundlagen der Kraftfahrzeugantriebe 90
330 Wahlmodul aus Master 91
10040 Bildsynthese 93
10080 Datenbanken und Informationssysteme 95
10120 Modellbildung und Simulation 97
10250 Parallele Systeme 98
14740 Partielle Differentialgleichungen (Modellierung, Analysis, Simulation) 99
29430 Computer Vision 101
29440 Geometric Modeling and Computer Animation 103
29450 Graphentheorie 105
29460 Algorithmen für die Kryptographie 106
29470 Machine Learning 107
29480 Loose Coupling and Message Based Applications 109
29510 Service Computing 111
29550 Algorithmische Geometrie 113
29570 Computer Interface Technologien 114
29580 Data Compression 115
29590 Digitale Systeme 116
29640 Mikrocontroller 117
29650 Parallele Programmierung 119
29660 Programmanalysen und Compilerbau 120
29670 Rapid Prototyping 122
29680 Real-Time Programming 123
29690 Real-Time Video Processing I 124
29710 Embedded Systems Engineering 125
29720 Mobile Computing 126
29730 Modelling, Simulation, and Specification 128
29740 Fachpraktikum Eingebettete Systeme 129
29750 Fachpraktikum Rechnerarchitektur 130
29760 Algorithmische Gruppentheorie 131
34940 Weiterführende Numerik partieller Differentialgleichungen 133
39250 Distributed Systems I 134
40680 Optimization 136
42420 High Performance Computing 138
42460 Numerische Simulation 140
42480 Ausgewählte Kapitel des Wissenschaftlichen Rechnens 141
42900 Business Process Management 142
42910 Advanced Business Process Management 144
42920 Hardware-Software-Codesign 145
45740 Rechnernetze II 146
45750 Fachpraktikum Verteilte Systeme 148
45760 Ausgewählte Kapitel der Algorithmentheorie 149
46660 Service Management and Cloud Computing, and Evaluation 150
48480 Data Engineering 151
48500 Image Synthesis 153
48550 Practical Course Information Systems 155
48560 Practical Course Robotics 156
48570 Practical Course Visual Computing 157
48580 Reinforcement Learning 158
48600 Robotics I 160
48620 Scientific Visualization 161
51540 Implementierung Finiter Elemente 162

51720 IT-Strategy 163
51740 Quantencomputing 165
55600 Advanced Information Management 166
55610 Information Integration 167
55620 Data Warehousing, Data Mining, and OLAP 169
55630 Information Visualization and Visual Analytics 171
55640 Correspondence Problems in Computer Vision 172
55650 Multimodal Interaction for Ubiquitous Computers 173
55740 Advanced Service Computing 174
56680 Automaten über unendlichen Objekten 176
56790 Parallele Numerik 178
56980 Fachpraktikum Cloud Architekturen und Management 180
57050 Compilerbau 181
57680 Einführung in die Chaostheorie 183
58190 Entwurf und Implementierung eines Compilers 185
58440 Fachpraktikum: Algorithmik 186
60120 Fachpraktikum Interaktive Systeme 187
60140 Sprachbau mit Language Workbenches 188
60860 3D Scanner - Algorithms and Systems 189
71740 System and Web Security 190
71760 Security and Privacy 191
71890 Mikrobiologische Diagnostik in der Humanmedizin 193
73600 Entwurf Robuster Systeme 194
73610 Hardwareorientierte Sicherheit 195
78900 Introduction to Modern Cryptography 196

400 Schlüsselqualifikationen fachaffin 198

38610 Seminar-INF 1 199
78610 Bachelor-Forschungsprojekt Informatik 201

81100 Bachelorarbeit Informatik 203

100 Basismodule

Zugeordnete Module:

10190	Mathematik für Informatiker und Softwaretechniker
10280	Programmierung und Software-Entwicklung
12060	Datenstrukturen und Algorithmen
36530	Rechnerorganisation 1
78620	Theoretische Informatik I
78630	Theoretische Informatik II
78650	Technische Grundlagen der Informatik

Modul: 10190 Mathematik für Informatiker und Softwaretechniker

2. Modulkürzel:	080300100	5. Moduldauer:	Zweisemestrig
3. Leistungspunkte:	18 LP	6. Turnus:	Wintersemester
4. SWS:	12	7. Sprache:	Deutsch

8. Modulverantwortlicher:	PD Dr. Andreas Markus Kollross
9. Dozenten:	Wolfgang Rump Andreas Markus Kollross Peter Lesky Wolf-Patrick Düll
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 Integralrechung (Funktionen einer und mehrerer Variablen, Ableitungen, Taylorentwicklungen, Extremwerte, Integration, Anwendungen) • Gewöhnliche Differentialgleichungen (elementar lösbare Differentialgleichungen, Existenz und Eindeutigkeit von Lösungen)
14. Literatur:	• Anna Sändig, Mathematik, Vorlesungskripte , SS 2007 • D. Hachenberger, Mathematik für Informatiker, 2005 • M. Brill, Mathematik für Informatiker, 2001 • P.Hartmann, Mathematik für Informatiker, 2002
15. Lehrveranstaltungen und -formen:	• 101902 Übung Mathematik 1 für Informatik und Softwaretechnik • 101901 Vorlesung 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), Schriftlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich

	Ein Übungsschein aus den beiden Veranstaltungen, jeweils im 1. oder 2. Fachsemester zu erwerben
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Geometrie

Modul: 10280 Programmierung und Software-Entwicklung

2. Modulkürzel:	051520005	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	9 LP	6. Turnus:	Wintersemester
4. SWS:	6	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Frank Leymann
9. Dozenten:	Frank Leymann
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 • Boolsche 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:	• 102802 Übung Programmierung und Softwareentwicklung • 102801 Vorlesung Programmierung und Softwareentwicklung
16. Abschätzung Arbeitsaufwand:	Präsenzstunden: 63 h Eigenstudiumstunden: 207 h Gesamtstunden: 270 h

17. Prüfungsnummer/n und -name:	• 10281 Programmierung und Software-Entwicklung (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [10281] Programmierung und Software-Entwicklung (PL), schriftliche Prüfung, 90 Min., Gewicht: 1.0, [Prüfungsvorleistung] 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 ... :	Datenstrukturen und Algorithmen
19. Medienform:	• Folien über Beamer • Tafelanschrieb
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 12060 Datenstrukturen und Algorithmen

2. Modulkürzel:	051510005	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	9 LP	6. Turnus:	Sommersemester
4. SWS:	6	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Andrés Bruhn
9. Dozenten:	Andrés Bruhn
10. Zuordnung zum Curriculum in diesem Studiengang:	
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. *Algorithmen und Datenstrukturen: Eine Einführung mit Java* . 5. Auflage, dpunkt-Verlag, 2013 • T. Ottmann, P. Widmayer. *Algorithmen und Datenstrukturen* . 5. Auflage, Springer-Verlag, 2012
15. Lehrveranstaltungen und -formen:	• 120602 Übung Datenstrukturen und Algorithmen • 120601 Vorlesung Datenstrukturen und Algorithmen
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 12061 Datenstrukturen und Algorithmen (PL), Schriftlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich Prüfungsvorleistung: Übungsschein. Die genauen Details der Übungsleistungen und Ihrer Anrechnung werden zu Beginn der Lehrveranstaltung bekannt gegeben.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Intelligente Systeme

Modul: 36530 Rechnerorganisation 1

2. Modulkürzel:	051700006	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Einführung in die Technische Informatik (14360) oder Technische Grundlagen der Informatik (78650)
12. Lernziele:	• Kenntnisse über die Grundlagen der Organisation von Rechnern und deren maschinennahe Programmierung • Grundzüge über die Beschreibung und den Entwurf von Hardwaresystemen
13. Inhalt:	Aufbau, Entwurf und maschinennahe Programmierung von Mikroprozessoren werden erläutert. In den Übungen wird das Wissen durch schriftliche Aufgaben sowie Labor-Ünbungen vertieft. Im Einzelnen werden behandelt: • Grundstrukturen: Logik Multiplexer, Speicher, Addierer, Multiplizierer, ALU • Zahlensysteme; Typ Interger, Typ Float, Gleitkommazahlen • Informationsdarstellung in Hardware: Daten- und Befehlsformate, Befehlssätze • Instruction Set Architecture • Assemblerprogrammierung • Grundzüge einer Hardware-Beschreibungssprache • RISC und CISC Architekturen • Steuerwerksentwurf und Mikroprogrammierung • Pipelining und Hazards • Speicherorganisation: Cachestrukturen und virtueller Speicher, Seitenverwaltung • Multi-Core Prozessoren • GPU-Architekturen und Programmierung • FPGA-Architekturen
14. Literatur:	Literatur, siehe Veranstaltungshinweise
15. Lehrveranstaltungen und -formen:	• 365301 Vorlesung Rechnerorganisation 1 • 365302 Praktische und theoretische Übungen Rechnerorganisation 1
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 36531 Rechnerorganisation 1 (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Parallele Systeme

Modul: 78620 Theoretische Informatik I

2. Modulkürzel:	050400101	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	apl. Prof. Dr. rer. nat. habil. Ulrich Hertrampf
9. Dozenten:	Volker Diekert Stefan Funke Ulrich Hertrampf
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	keine
12. Lernziele:	Beherrschung wichtiger theoretischer Grundlagen der Informatik, insbesondere die Theorie und Algorithmik endlicher Automaten. Kennenlernen, Einordnung und Trennung der Chomskyschen Sprachklassen.
13. Inhalt:	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:	Uwe Schöning, Theoretische Informatik - kurzgefasst, 1999.
15. Lehrveranstaltungen und -formen:	• 786201 Vorlesung/Übung Theoretische Informatik I
16. Abschätzung Arbeitsaufwand:	Vorlesung und Übung Theoretische Informatik I
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), • 78621 Theoretische Informatik I (PL), Schriftlich, Gewichtung: 1 USL-V (Übungsschein als Vorleistung) Schriftliche Prüfung, 120 Minuten
18. Grundlage für ... :	Theoretische Informati II und Theoretische Informatik III
19. Medienform:	
20. Angeboten von:	Theoretische Informatik

Modul: 78630 Theoretische Informatik II

2. Modulkürzel:	050400102	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	apl. Prof. Dr. rer. nat. habil. Ulrich Hertrampf
9. Dozenten:	Volker Diekert Stefan Funke Ulrich Hertrampf
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Theoretische Informatik I
12. Lernziele:	Die Teilnehmer beherrschen wichtige theoretische Grundlagen der Informatik, können Probleme in Kategorien einordnen wie entscheidbar/unentscheidbar, effizient lösbar, deterministische/ nichtdeterministische Berechnungen. Sie kennen die Grundbegriffe der Komplexitätstheorie und die wichtigsten Komplexitätsklassen.
13. Inhalt:	Gleichwertigkeit der verschiedenden Konkretisierungen des Algorithmenbegriffs, Churchsche 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:	Volker Diekert, Komplexitätstheorie (Vorlesungsskript), 2007.
15. Lehrveranstaltungen und -formen:	• 786301 Vorlesung/Übung Theoretische Informatik II
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), • 78631 Theoretische Informatik II (PL), Schriftlich, Gewichtung: 1 USL-V (Übungsschein als Vorleistung) Schriftliche Prüfung, 120 Minuten
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Theoretische Informatik

Modul: 78650 Technische Grundlagen der Informatik

2. Modulkürzel:	051711666	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Martin Radetzki
9. Dozenten:	Martin Radetzki
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Keine
12. Lernziele:	Grundlegendes Verständnis elektrischer Schaltkreise und der Funktionsweise der Bauelemente und Komponenten von Computersystemen, wie Transistoren, digitale Halbleiterschaltungen, Speicher. Fähigkeit zur Analyse, Konstruktion und Optimierung digitaler Schaltungen von begrenzter Komplexität.
13. Inhalt:	• Informationsbegriff, Codierung, Darstellung mit analogen Größen • Übersicht über den Entwurf informationsverarbeitender Systeme • Boole'sche Algebra und Schaltalgebra • Schaltnetze / kombinatorische Netzwerke • Elektrostatisches Feld, Potential, Spannung und Kondensator • Elektrischer Strom, elektrische Netzwerke und Widerstand • Halbleitertechnik, Diode, Transistor • Digitale Grundschaltungen, CMOS Grund- und Komplexgatter • Optimierung kombinatorischer Schaltungen • Verzögerungsanalyse • Kombinatorische Komponenten von Rechensystemen • Sequentielle Komponenten von Rechensystemen • Modelle sequentiellen Verhaltens, Schaltwerke / sequentielle Netzwerke • Entwurfsmethodik und Entwurfsautomatisierung
14. Literatur:	• Manfred Albach, Grundlagen der Elektrotechnik I, 2004, Pearson Studium • Dirk Hoffmann: Grundlagen der Technischen Informatik, 2. Auflage 2010, Hanser • Wolfram Schiffmann, Robert Schmitz: Technische Informatik 1, 5. Auflage 2004, Springer • Randy Katz: Contemporary Logic Design, 2. Auflage 2005, Pearson Education • Bernd Becker, Rolf Drechsler, Paul Molitor: Technische Informatik, 1. Auflage 2005, Pearson Studium • Hans Liebig: Logischer Entwurf digitaler Systeme, 4. Auflage 2006, Springer
15. Lehrveranstaltungen und -formen:	• 786501 Vorlesung Technische Grundlagen der Informatik 3,0 SWS • 786502 Übung Technische Grundlagen der Informatik 1,0 SWS
16. Abschätzung Arbeitsaufwand:	Vorlesung Technische Grundlagen der Informatik; 3,0 SWS Übung Technische Grundlagen der Informatik; 1,0 SWS
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V),

	• 78651 Technische Grundlagen der Informatik (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 Prüfungsvorleistung (USL-V), schriftlich, eventuell mündlich: Als Zulassungsvoraussetzung zur Klausur ist folgende Vorleistung zu erbringen: aktive und erfolgreiche Teilnahme an einer Mindestanzahl der Übungen, Präsentation der Lösung wenigstens einer Aufgabe, Bestehen eines Zulassungstests. Details werden zu Beginn der Lehrveranstaltung bekanntgegeben. Technische Grundlagen der Informatik (PL), schriftliche Prüfung, 120 Min., Gewicht: 1.0
18. Grundlage für ... :	Rechnerorganisation 1
19. Medienform:	
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)

200 Kernmodule

Zugeordnete Module:
- 210 Pflichtmodule
- 220 Wahlpflichtmodule

210 Pflichtmodule

Zugeordnete Module:

78660	Programmierprojekt
78670	Numerische Grundlagen
78680	Statistische und stochastische Grundlagen
78730	Theoretische Informatik III
78740	Wissenschaftliche Methoden in der Informatik

Modul: 78660 Programmierprojekt

2. Modulkürzel:	050410103	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester/ Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Funke
9. Dozenten:	Stefan Funke Stefan Wagner
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Programmierung und Softwareentwicklung • Datenstrukturen und Algorithmen
12. Lernziele:	Die Studierenden können eine größere Software in einem kleinen Team entwickeln.
13. Inhalt:	Die Studierende lernen verschiedene Konzepte zur Programmierung von Softwaresystemen kennen und üben diese ein. Dazu bearbeiten sie mehrere kleinere Übungsaufgaben zum Kennenlernen der Technologie sowie eine größere Projektarbeit im Team. Die konkreten verwendeten Technologien werden von den jeweiligen Dozenten festgelegt.
14. Literatur:	Jochen Ludewig, Horst Lichter. Software Engineering. Grundlagen, Menschen, Prozesse, Techniken. 2007
15. Lehrveranstaltungen und -formen:	• 786601 Programmierprojekt, Praktikum, 4,0 SWS
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	78661 Programmierprojekt (USL), Sonstige, Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	Angaben zu verwendeten Medien (Tafel, Flipchart etc.)
20. Angeboten von:	Software Engineering

Modul: 78670 Numerische Grundlagen

2. Modulkürzel:	051240100	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Miriam Mehl
9. Dozenten:	Dirk Pflüger Miriam Mehl Stefan Zimmer
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Modul 10190 Mathematik für Informatiker und Softwaretechniker
12. Lernziele:	Beherrschung grundlegender Begriffe und Methoden der Numerik, Kenntnis der Anwendungsbereiche und Gültigkeitsgrenzen der erlernten Methoden, Fähigkeit zur Quantifizierung der Auswirkungen von Näherungen und zur Beurteilung der wesentlichen Vor- und Nachteile der vorgestellten Methoden, Fähigkeit zur grundlegenden Analyse und Überprüfung der Ergebnisse numerischer Verfahren
13. Inhalt:	In Ergänzung der Mathematik-Grundausbildung vermittelt diese Vorlesung folgende Grundkenntnisse: • Numerische Algorithmik • Gleitpunktzahlen und Gleitpunkarithmetik • Lineare Gleichungssysteme • nichtlineare Gleichungen • Interpolation und Approximation • Integration • Gewöhnliche Differentialgleichungen Dabei wird ein konstruktiv-algorithmischer Zugang gewählt, der sich an konkreten Aufgabenstellungen aus der Informatik orientiert.
14. Literatur:	• Dahmen, Reusken, Numerik für Ingenieure • Gander, Gander, Kwok: Scientific Computing, Kapitel 1-5 • Schwarz, Köckler, Numerische Mathematik • Huckle, Schneider, Numerik für Informatiker • Skript
15. Lehrveranstaltungen und -formen:	• 786701 Vorlesung Numerische Grundlagen • 786702 Übung Numerische Grundlagen
16. Abschätzung Arbeitsaufwand:	Numerische Grundlagen, Vorlesung, 3,0 SWS Numerische Grundlagen, Übung, 1,0 SWS
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), • 78671 Numerische Grundlagen (PL), Schriftlich, Gewichtung: 1 Unbenotete Studienleistung als Vorleistung (USL-V): schriftlich, eventuell mündlich Numerische Grundlagen (PL), schriftliche Prüfung,90 Min. , Gewicht: 1.0, Prüfungsvorleistung: Übungsschein (USL-V)

18. Grundlage für ... :	Grundlagen des wissenschaftlichen Rechnens (42410)
19. Medienform:	Tafel, Beamer
20. Angeboten von:	Simulation großer Systeme

Modul: 78680 Statistische und stochastische Grundlagen

2. Modulkürzel:	051240090	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Miriam Mehl
9. Dozenten:	Dirk Pflüger Miriam Mehl Stefan Zimmer
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	*Modul 10190 Mathematik für Informatiker und Softwaretechniker*
12. Lernziele:	*Beherrschung grundlegender Begriffe der Stochastik und Statistik, Kenntnis der Anwendungsbereiche und Gültigkeitsgrenzen stochastischer Modelle, Kenntnis und Fähigkeit zur Verwendung stochastischer Fehlermodelle und Konvergenzbegriffe, Beherrschung der Modellierung einfacher Probleme und des Entwurfs von Tests mit stochastischen Methoden*
13. Inhalt:	*In Ergänzung der Mathematik-Grundausbildung vermittelt diese Vorlesung folgende Grundkenntnisse:* • *Endliche, diskrete und allgemeine Wahrscheinlichkeitsräume* • *Beispiele für diskrete und stetige Verteilungen* • *Grenzwertsätze* • *Elementare induktive Statistik* • *Methoden und Algorithmen der Datenanalyse* • *einfache Testmethoden* • *Stochastische Prozesse* *Dabei wird ein konstruktiv-algorithmischer Zugang gewählt, der sich an konkreten Aufgabenstellungen aus der Informatik orientiert.*
14. Literatur:	• *Henze, Stochastik für Einsteiger* • *Schickinger, Steger, Diskrete Strukturen, Band 2* • *Fahrmeir et.al., Statistik - der Weg zur Datenanalyse* • *Skript*
15. Lehrveranstaltungen und -formen:	• 786801 Vorlesung Statistische und stochastische Grundlagen • 786802 Übung Statistische und stochastische Grundlagen
16. Abschätzung Arbeitsaufwand:	*Tafel, Beamer*
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), • 78681 Statistische und stochastische Grundlagen (PL), Schriftlich, Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Simulation großer Systeme

Modul: 78730 Theoretische Informatik III

2. Modulkürzel:	050400103	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	apl. Prof. Dr. rer. nat. habil. Ulrich Hertrampf
9. Dozenten:	Volker Diekert Stefan Funke Ulrich Hertrampf
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Theoretische Informatik I,II
12. Lernziele:	Die Studierenden kennen wichtige Entwurfsstrategien und Analysemethoden. Sie haben Grundkenntnisse der Kombinatorik, die in vielen Bereichen der Informatik benötigt werden.
13. Inhalt:	Entwurfsstrategien: Teile und Herrsche, gierig (greedy), Dynamisches Programmieren, Randomisierte Algorithmen, Graphen, elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, RSA-Verfahren.
14. Literatur:	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms (Second Edition), 2001.
15. Lehrveranstaltungen und -formen:	• 787301 Vorlesung/Übung Theoretische Informatik III
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), • 78731 Theoretische Informatik III (PL), Schriftlich, Gewichtung: 1 USL-V (Übungsschein als Vorleistung) Schriftliche Prüfung, 120 Minuten
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Theoretische Informatik

Modul: 78740 Wissenschaftliche Methoden in der Informatik

2. Modulkürzel:	051520021	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	3 LP	6. Turnus:	Wintersemester
4. SWS:	2	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Wagner
9. Dozenten:	Stefan Wagner
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Vorher oder parallel sollten die Statistischen und stochastischen Grundlagen gehört werden.
12. Lernziele:	Die Studierenden haben ein grundlegendes Verständnis von Wissenschaftsphilosophie und der wissenschaftlichen Methode. Sie können mit wissenschaftlicher Literatur umgehen und erste wissenschaftliche Aufsätze und Vorträge verfassen. Sie können in einfachen Kontexten empirische Methoden für Informatikfragestellungen einsetzen.
13. Inhalt:	• Wissenschaftsphilosophie und wissenschaftliche Methode • Literaturarbeit, Referenzieren, wissenschaftliches Schreiben und Präsentieren • Empirische Methoden in der Informatik • Anwendung deskriptiver Statistik und statistischer Tests • Ethische Richtlinien bei der Durchführung von Studien
14. Literatur:	• Leedy, Ormrod. Practical Research: Planning and Design. Pearson Prentice Hall, 2009 • Field, Andy P., and Graham Hole. How to design and report experiments. London: Sage Publications, 2003. • Fink, Arlene, ed. How to conduct surveys: A step-by-step guide. Sage, 2009.
15. Lehrveranstaltungen und -formen:	• 787401 Vorlesung Wissenschaftliche Methoden in der Informatik • 787402 Übung Wissenschaftliche Methoden in der Informatik
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	78741 Wissenschaftliche Methoden in der Informatik (USL), Schriftlich oder Mündlich, Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Software Engineering

220 Wahlpflichtmodule

Zugeordnete Module:
2201 Praktische Informatik
2202 Wahlpflichtmodule allgemein

2201 Praktische Informatik

Zugeordnete Module:
- 10210 Mensch-Computer-Interaktion
- 10220 Modellierung
- 36100 Programmierparadigmen
- 40090 Systemkonzepte und -programmierung

Modul: 10210 Mensch-Computer-Interaktion

2. Modulkürzel:	051900001	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Jun.-Prof. Dr. Niels Henze
9. Dozenten:	Niels Henze Tonja Machulla
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 Dachselt. Interaktive Systeme 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung. Springer, Berlin, 2. Auflage. 2010 • Alan Dix, Janet Finley, Gregory Abowd, Russell Beale, Human-Computer Interaction, 2004 • Ben Shneiderman, Catherine Plaisant, Designing the User Interfaces, 2005
15. Lehrveranstaltungen und -formen:	• 102101 Vorlesung Mensch-Computer-Interaktion • 102102 Übung Mensch-Computer-Interaktion
16. Abschätzung Arbeitsaufwand:	

17. Prüfungsnummer/n und -name:	• 10211 Mensch-Computer-Interaktion (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Soziokognitive Systeme

Modul: 10220 Modellierung

2. Modulkürzel:	052010001	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann
9. Dozenten:	Bernhard Mitschang Frank Leymann Uwe Breitenbücher
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Modul 10280 Programmierung und Software-Entwicklung • Modul 12060 Datenstrukturen und Algorithmen • Modul 40090 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 und komplexe Objekte • Relationenmodell und Relationenalgebra , Überblick SQL - Transformationen von ER nach Relationen, Normalisierung • XML, DTD, XML-Schema, Info-Set, Namensräume • Metamodelle und Repository - RDF, RDF-S und 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 und Design, 2nd Edition, 1994. • H.J. Habermann, F. Leymann, Repository , Oldenbourg 1993. • W. Reisig, Petri-Netze , Vieweg und Teubner 2010. • B. Silver, BPMN Method und Style ,Cody-Cassidy Press 2009.
15. Lehrveranstaltungen und -formen:	• 102202 Übung Modellierung • 102201 Vorlesung Modellierung
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 10221 Modellierung (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [10221] Modellierung (PL), schriftliche Prüfung, 90 Min., Gewicht: 1.0, Prüfungsvorleistung: Übungsschein [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich

18. Grundlage für ... :	Architektur von Anwendungssystemen Datenbanken und Informationssysteme
19. Medienform:	
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 36100 Programmierparadigmen

2. Modulkürzel:	051510010	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Erhard Plödereder
9. Dozenten:	Erhard Plödereder
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Programmiererfahrung in mindestens einer Programmiersprache, vornehmlich Java, so wie z. B. im Modul "Programmierung und Software Entwicklung"(10280) erworben.
12. Lernziele:	Die Studierenden haben grundlegende Konzepte von Programmiersprachen verstanden, die dem Erlernen weiterer Sprachen und dem vertieften Verständnis ihnen bekannter Sprachen dienlich sind. Sie haben deren Anwendung in mindestens einer weiteren Programmiersprache ihrer Ausbildung verstanden. Sie können ihre Kenntnisse in einfachen Programmen anwenden. Sie können weitere Programmiersprachen in ihrer akademischen und beruflichen Karriere schneller und präziser erlernen.
13. Inhalt:	Überblick typischer Konzepte in Programmiersprachen und ihrer Auswirkungen auf die Sprache und deren Anwendung: • Grundsätzliche Ausführungsmodelle • Speichermodelle und deren Konsequenzen • Datentypen und Typsysteme • unterschiedliche Bindungskonzepte und ihre Auswirkungen • objekt-orientierte Sprachkonzepte • Abstraktion und Kompositionsmechanismen • funktionale Sprachen. • Eventuell werden auch Elemente der parallelen Programmierung und der Logik-Programmierung mit einbezogen. Die Vorlesung ist kein Streifzug durch diverse Programmiersprachen, sondern die Vorstellung zugrundeliegender Prinzipien, und ihrer Begründung aus der Sicht des Software Engineering, insbesondere der Zuverlässigkeit der Anwendung, und, wo nötig, der Implementierungsmodelle.
14. Literatur:	• Sebesta, Robert, Concepts of Programming Languages, Pearson Verlag, 2010 (Hörerschein verfügbar). • weitere Literatur wird zu Beginn der Lehrveranstaltung und auf den Webseiten bekanntgegeben.
15. Lehrveranstaltungen und -formen:	• 361001 Vorlesung Programmierparadigmen • 361002 Übung Programmierparadigmen
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 36101 Programmierparadigmen (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich

	[36101] Programmierparadigmen (PL), schriftliche Prüfung, 90 Min., Gewicht: 1.0 [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Programmiersprachen und Übersetzerbau

Modul: 40090 Systemkonzepte und -programmierung

2. Modulkürzel:	051200005	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Kurt Rothermel
9. Dozenten:	Kurt Rothermel Frank Dürr
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Modul 10280 Programmierung und Software-Entwicklung • Modul 12060 Datenstrukturen und Algorithmen
12. Lernziele:	• Verstehen grundlegender Architekturen und Organisationsformen von Software-Systemen • Verstehen systemnaher Konzepte und Mechanismen • Kann existierende Systemplattformen und Betriebssysteme hinsichtlich ihrer Eigenschaften analysieren und anwenden. • Kann systemnahe Software entwerfen und implementieren. • Kann nebenläufige Programme entwickeln • Kann mit Experten anderer Fachgebiete die Anwendung von Systemfunktionen abstimmen.
13. Inhalt:	Grundlegende Systemstrukturen - und organisationen • Multitaskingsystem • Multiprozessorsystem • Verteiltes System Modellierung und Analyse nebenläufiger Programme • Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm • Korrektheit- und Leitungskriterien Betriebssystemkonzepte • Organisation von Betriebssystemen • Prozesse und Threads • Eingabe/Ausgabe • Scheduling Konzepte zur Synchronisation über gemeinsamen Speicher • Synchronisationsprobleme und -lösungen • Synchronisationswerkzeuge: Semaphor, Monitor Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer • Taxonomie: Kommunikation und Synchronisation • Nachrichten als Kommunikationskonzept • Höhere Kommunikationskonzepte Basisalgorithmen für Verteilte Systeme • Erkennung globaler Eigenschaften • Schnappschussproblem • Konsistenter globaler Zustand • Verteilte Terminierung Praktische nebenläufige Programmierung in Java • Threads und Synchronisation • Socketschnittstelle • RMI Programmierung
14. Literatur:	Literatur, siehe Webseite zur Veranstaltung

15. Lehrveranstaltungen und -formen:	• 400901 Vorlesung Systemkonzepte und -programmierung • 400902 Übung Systemkonzepte und -programmierung
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 40091 Systemkonzepte und -programmierung (PL), Schriftlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [40091] Systemkonzepte und -programmierung (PL), schriftliche Prüfung, 120 Min., Gewicht: 1.0 [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Verteilte Systeme

2202 Wahlpflichtmodule allgemein

Zugeordnete Module:

- 10060 Computergraphik
- 10170 Imaging Science
- 10210 Mensch-Computer-Interaktion
- 10220 Modellierung
- 14390 Programmentwicklung
- 16500 Software Engineering
- 17210 Einführung in die Softwaretechnik
- 36100 Programmierparadigmen
- 40090 Systemkonzepte und -programmierung
- 42410 Grundlagen des Wissenschaftlichen Rechnens
- 78640 Grundlagen der Informationssicherheit
- 78750 Rechnerorganisation 2

Modul: 10060 Computergraphik

2. Modulkürzel:	051900002	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Thomas Ertl
9. Dozenten:	Thomas Ertl Daniel Weiskopf Michael Krone Guido Reina
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 Verdeckungsberechung • 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:	• 100602 Übung Computergraphik • 100601 Vorlesung Computergraphik
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 10061 Computergraphik (PL), Schriftlich, 60 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich Prüfungsvorleistung: Übungsschein.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Praktische Informatik (Dialogsysteme)

Modul: 10170 Imaging Science

2. Modulkürzel:	051900210	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Andrés Bruhn
9. Dozenten:	Andrés Bruhn
10. Zuordnung zum Curriculum in diesem Studiengang:	
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.
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
14. Literatur:	• Bässmann, Henning, Kreyss, 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:	• 101702 Übung Imaging Science • 101701 Vorlesung Imaging Science
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 10171 Imaging Science (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [10171] Imaging Science (PL), schriftlich, eventuell mündlich, 120 Min., Gewicht: 1.0,

	Prüfungsvorleistung (USL-V): Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben, schriftlich, eventuell mündlich
18. Grundlage für ... :	Modul Computer VisionModul Correspondence Problems in Computer Vision
19. Medienform:	
20. Angeboten von:	Intelligente Systeme

Modul: 10210 Mensch-Computer-Interaktion

2. Modulkürzel:	051900001	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Jun.-Prof. Dr. Niels Henze
9. Dozenten:	Niels Henze Tonja Machulla
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 Dachselt. Interaktive Systeme 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung. Springer, Berlin, 2. Auflage. 2010 • Alan Dix, Janet Finley, Gregory Abowd, Russell Beale, Human-Computer Interaction, 2004 • Ben Shneiderman, Catherine Plaisant, Designing the User Interfaces, 2005
15. Lehrveranstaltungen und -formen:	• 102101 Vorlesung Mensch-Computer-Interaktion • 102102 Übung Mensch-Computer-Interaktion
16. Abschätzung Arbeitsaufwand:	

17. Prüfungsnummer/n und -name:	• 10211 Mensch-Computer-Interaktion (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Soziokognitive Systeme

Modul: 10220 Modellierung

2. Modulkürzel:	052010001	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann
9. Dozenten:	Bernhard Mitschang Frank Leymann Uwe Breitenbücher
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Modul 10280 Programmierung und Software-Entwicklung • Modul 12060 Datenstrukturen und Algorithmen • Modul 40090 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 und komplexe Objekte • Relationenmodell und Relationenalgebra , Überblick SQL - Transformationen von ER nach Relationen, Normalisierung • XML, DTD, XML-Schema, Info-Set, Namensräume • Metamodelle und Repository - RDF, RDF-S und 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 und Design, 2nd Edition, 1994. • H.J. Habermann, F. Leymann, Repository , Oldenbourg 1993. • W. Reisig, Petri-Netze , Vieweg und Teubner 2010. • B. Silver, BPMN Method und Style ,Cody-Cassidy Press 2009.
15. Lehrveranstaltungen und -formen:	• 102202 Übung Modellierung • 102201 Vorlesung Modellierung
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 10221 Modellierung (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [10221] Modellierung (PL), schriftliche Prüfung, 90 Min., Gewicht: 1.0, Prüfungsvorleistung: Übungsschein [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich

18. Grundlage für ... :	Architektur von Anwendungssystemen Datenbanken und Informationssysteme
19. Medienform:	
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 14390 Programmentwicklung

2. Modulkürzel:	051520120	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Wagner
9. Dozenten:	Stefan Wagner Jan-Peter Ostberg
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Programmierung und Softwareentwicklung • Einführung in die Softwaretechnik
12. Lernziele:	Die Teilnehmer kennen die Prinzipien der objektorientierten Programmierung und sind in der Lage, Programme in UML zu beschreiben und in Java zu implementieren.
13. Inhalt:	• Grundlagen der objektorientierten Programmierung • Spezifikation und Entwurf objektorientierter Programme mit UML • Vertiefte Programmierung in Java
14. Literatur:	• Rumbaugh, Jacobson, Booch, The unified modeling language reference manual, 2nd ed., 2004 • Rupp, Queins, Zengler, UML 2 glasklar: Praxiswissen für die UML-Modellierung, 3. Aufl. 2007 • Ullenboom: Java ist auch eine Insel. Galileo Computing, 8. Aufl. 2009
15. Lehrveranstaltungen und -formen:	• 143901 Vorlesung Programmentwicklung • 143902 Übung Programmentwicklung
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	14391 Programmentwicklung (PL), Schriftlich, 60 Min., Gewichtung: 1 Klausur 60 min, keine Vorleistungen.
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: 16500 Software Engineering

2. Modulkürzel:	051520110	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Steffen Becker
9. Dozenten:	Steffen Becker André van Hoorn
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Einführung in die Softwaretechnik • Programmierung und Softwarentwicklung
12. Lernziele:	Die Teilnehmer haben tiefe und umfassende Kenntnisse auf dem Gebiet des Software Engineerings.
13. Inhalt:	Ergänzend zur Einführung in die Softwaretechnik und daran anknüpfend, behandelt diese Lehrveranstaltung folgende Themen: • Organisationsaspekte des Software Engineering • Softwareentwicklungsprozesse, Prozessbewertung und -verbesserung • Anforderungsanalyse • Softwarearchitektur • Realisierung und Debugging • Softwarequalitätssicherung • Softwarewartung • Model-Driven Software Development • Weitere ausgewählte Kapitel des Software Engineerings
14. Literatur:	• Summerville, Software Engineering, AW • Ludewig J., Lichter, H., Software Engineering - Grundlagen, Menschen, Prozesse, Techniken, 2. Aufl. 2010 • Liggesmeyer P., Software-Qualität. Testen, Analysieren und Verifizieren von Software. Spektrum Akademischer Verlag, 2002.
15. Lehrveranstaltungen und -formen:	• 165002 Übung Software Engineering • 165001 Vorlesung Software Engineering
16. Abschätzung Arbeitsaufwand:	Vorleszungen und Zentralübungen
17. Prüfungsnummer/n und -name:	16501 Software Engineering (PL), Schriftlich, 90 Min., Gewichtung: 1 90min Klausur
18. Grundlage für ... :	Requirements Engineering and Software ArchitectureModel-Driven Software DevelopmentSoftware Qualität und -Wartung
19. Medienform:	Folien, Videoaufzeichnung, Übungsblätter, elektronische Umfragen und -Tests
20. Angeboten von:	Zuverlässige Software-Systeme

Modul: 17210 Einführung in die Softwaretechnik

2. Modulkürzel:	051520015	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Wagner
9. Dozenten:	Stefan Wagner
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	- Modul 10280 Programmierung und Software-Entwicklung - Modul 12060 Datenstrukturen und Algorithmen - sowie entsprechende Programmiererfahrung
12. Lernziele:	Die Veranstaltung liefert einen ersten Einblick in die Softwaretechnik. Sie ist abgestimmt auf die Software-Qualität im 1. und Programmentwicklung im 3. Semester. Die Teilnehmer kennen die Grundbegriffe der Softwaretechnik und haben wichtige Techniken des Softwareprojekt-Managements und der Software-Entwicklung erlernt. Sie kennen Scrum als eine konkrete Vorgehensweise zur Softwareentwicklung
13. Inhalt:	Die Vorlesung behandelt technische und andere Aspekte der Softwarebearbeitung, wie sie in der Praxis stattfindet. Die einzelnen Themen sind: • Abgrenzung und Motivation des Software Engineerings • Vorgehensmodelle, agiles Vorgehen, Scrum • Software-Management • Software-Prüfung und Qualitätssicherung • Methoden, Sprachen und Werkzeuge für die einzelnen Phasen:Spezifikation, Grobentwurf, Feinentwurf, Implementierung, Test
14. Literatur:	• Ludewig, Lichter: Software Engineering. dpunkt-Verlag, Heidelberg. 2. Aufl. 2010 • Pfleeger, Atlee: Software Engineering. Pearson, 2010 • Rubin: Essential Scrum. Addison-Wesley, 2013
15. Lehrveranstaltungen und -formen:	• 172101 Vorlesung Einführung in die Softwaretechnik • 172102 Übung Einführung in die Softwaretechnik
16. Abschätzung Arbeitsaufwand:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	• 17211 Einführung in die Softwaretechnik (PL), Schriftlich, 60 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich, 30 Min. [17211] Einführung in die Softwaretechnik (PL), schriftliche Prüfung, 60 Min., Gewicht: 1.0, [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich, Hausaufgaben
18. Grundlage für ... :	- Modul Software Engineering - Modul Software-Praktikum
19. Medienform:	• Folien am Beamer unterstützt durch Tafel und Overhead • Dokumente, Links und Diskussionsforum in ILIAS

20. Angeboten von: Software Engineering

Modul: 36100 Programmierparadigmen

2. Modulkürzel:	051510010	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Erhard Plödereder
9. Dozenten:	Erhard Plödereder
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Programmiererfahrung in mindestens einer Programmiersprache, vornehmlich Java, so wie z. B. im Modul "Programmierung und Software Entwicklung"(10280) erworben.
12. Lernziele:	Die Studierenden haben grundlegende Konzepte von Programmiersprachen verstanden, die dem Erlernen weiterer Sprachen und dem vertieften Verständnis ihnen bekannter Sprachen dienlich sind. Sie haben deren Anwendung in mindestens einer weiteren Programmiersprache ihrer Ausbildung verstanden. Sie können ihre Kenntnisse in einfachen Programmen anwenden. Sie können weitere Programmiersprachen in ihrer akademischen und beruflichen Karriere schneller und präziser erlernen.
13. Inhalt:	Überblick typischer Konzepte in Programmiersprachen und ihrer Auswirkungen auf die Sprache und deren Anwendung: • Grundsätzliche Ausführungsmodelle • Speichermodelle und deren Konsequenzen • Datentypen und Typsysteme • unterschiedliche Bindungskonzepte und ihre Auswirkungen • objekt-orientierte Sprachkonzepte • Abstraktion und Kompositionsmechanismen • funktionale Sprachen. • Eventuell werden auch Elemente der parallelen Programmierung und der Logik-Programmierung mit einbezogen. Die Vorlesung ist kein Streifzug durch diverse Programmiersprachen, sondern die Vorstellung zugrundeliegender Prinzipien, und ihrer Begründung aus der Sicht des Software Engineering, insbesondere der Zuverlässigkeit der Anwendung, und, wo nötig, der Implementierungsmodelle.
14. Literatur:	• Sebesta, Robert, Concepts of Programming Languages, Pearson Verlag, 2010 (Hörerschein verfügbar). • weitere Literatur wird zu Beginn der Lehrveranstaltung und auf den Webseiten bekanntgegeben.
15. Lehrveranstaltungen und -formen:	• 361001 Vorlesung Programmierparadigmen • 361002 Übung Programmierparadigmen
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 36101 Programmierparadigmen (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich

	[36101] Programmierparadigmen (PL), schriftliche Prüfung, 90 Min., Gewicht: 1.0 [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Programmiersprachen und Übersetzerbau

Modul: 40090 Systemkonzepte und -programmierung

2. Modulkürzel:	051200005	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Kurt Rothermel
9. Dozenten:	Kurt Rothermel Frank Dürr
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Modul 10280 Programmierung und Software-Entwicklung • Modul 12060 Datenstrukturen und Algorithmen
12. Lernziele:	• Verstehen grundlegender Architekturen und Organisationsformen von Software-Systemen • Verstehen systemnaher Konzepte und Mechanismen • Kann existierende Systemplattformen und Betriebssysteme hinsichtlich ihrer Eigenschaften analysieren und anwenden. • Kann systemnahe Software entwerfen und implementieren. • Kann nebenläufige Programme entwickeln • Kann mit Experten anderer Fachgebiete die Anwendung von Systemfunktionen abstimmen.
13. Inhalt:	Grundlegende Systemstrukturen - und organisationen • Multitaskingsystem • Multiprozessorsystem • Verteiltes System Modellierung und Analyse nebenläufiger Programme • Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm • Korrektheit- und Leitungskriterien Betriebssystemkonzepte • Organisation von Betriebssystemen • Prozesse und Threads • Eingabe/Ausgabe • Scheduling Konzepte zur Synchronisation über gemeinsamen Speicher • Synchronisationsprobleme und -lösungen • Synchronisationswerkzeuge: Semaphor, Monitor Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer • Taxonomie: Kommunikation und Synchronisation • Nachrichten als Kommunikationskonzept • Höhere Kommunikationskonzepte Basisalgorithmen für Verteilte Systeme • Erkennung globaler Eigenschaften • Schnappschussproblem • Konsistenter globaler Zustand • Verteilte Terminierung Praktische nebenläufige Programmierung in Java • Threads und Synchronisation • Socketschnittstelle • RMI Programmierung
14. Literatur:	Literatur, siehe Webseite zur Veranstaltung

15. Lehrveranstaltungen und -formen:	• 400901 Vorlesung Systemkonzepte und -programmierung • 400902 Übung Systemkonzepte und -programmierung
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 40091 Systemkonzepte und -programmierung (PL), Schriftlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [40091] Systemkonzepte und -programmierung (PL), schriftliche Prüfung, 120 Min., Gewicht: 1.0 [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Verteilte Systeme

Modul: 42410 Grundlagen des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240020	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Dirk Pflüger
9. Dozenten:	Miriam Mehl Stefan Zimmer Dirk Pflüger
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	42411 Grundlagen des Wissenschaftlichen Rechnens (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1 [42411] Grundlagen des Wissenschaftlichen Rechnens (PL), schriftlich oder mündlich, 90 Min., Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Simulation Software Engineering

Modul: 78640 Grundlagen der Informationssicherheit

2. Modulkürzel:	052900001	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. rer. nat. Ralf Küsters
9. Dozenten:	Ralf Küsters
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Empfohlen werden Kenntnisse in den Grundlagen der Informatik und Mathematik wie sie in den ersten beiden Semestern eines Bachelorstudiengangs Informatik/Mathematik vermittelt werden.
12. Lernziele:	Die Veranstaltung soll Studierende einerseits einen Überblick über die Informationssicherheit vermitteln und sie für dieses Thema sensibilisieren. Zum anderen lernen Studierende grundlegende Konzepte der Informationssicherheit kennen.

13. Inhalt:	Die moderne digitale Gesellschaft ist ohne Informationssicherheit nicht denkbar. Daten und sogar Geld sind digital, Kommunikation ist digital, kritische Infrastrukturen (Banken, Industrieanlagen, Verkehrsmittel, etc.) hängen stark von IT-Systemen ab. Es gibt kaum Lebensbereiche, die nicht von der Digitalisierung durchdrungen sind. Die digitale Welt ist deshalb ein attraktives Ziel für Angreifer aller Art (Kriminelle, Geheimdienste, Industriespione, Staaten, etc.) und mittlerweile ständigen Angriffen ausgesetzt. Diese Veranstaltung bietet eine erste Einführung in die Informationssicherheit. Unter anderem werden folgende Themen behandelt, wobei jeweils in der Praxis eingesetze Verfahren betrachtet werden, deren Sicherheit untersucht wird und bekannte Angriffe vorgestellt werden.
	• Grundlagen der Kryptographie
	• (Un-)Sicherheit von Netzwerkprotokollen, wie TCP, DNS, BGP, einschließlich
	  • Denial-of-Service-Angriffe
	  • Firewalls
	• Kryptographische Protokolle (TLS, SSH, WPA2, etc.)
	  • Zertifikate und Public-Key-Infrastrukturen
	  • Authentifizierung und Schlüsselaustausch
	• Zugriffskontrolle, z.B. in Linux, SELinux und Android
	• Grundlagen der Websicherheit
	• Ausblick Systemsicherheit

14. Literatur:	Wird in der Veranstaltung bekanntgegeben.
15. Lehrveranstaltungen und -formen:	• 786401 Vorlesung/Übung zu Grundlagen der Informationssicherheit
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), • 78641 Grundlagen der Informationssicherheit (PL), Schriftlich oder Mündlich, Gewichtung: 1 Unbenotete Studienleistung als Vorleistung (USL-V); ausreichende Punktzahl in den Übungen sowie ggf. in einer Zwischenklausur;

	Prüfungsleistung (PL): Klausur (90 Minuten) oder mündliche Prüfung (30 Minuten) zur Vorlesung und Übung Grundlagen der Informationssicherheit
18. Grundlage für ... :	
19. Medienform:	Projector, blackboard
20. Angeboten von:	Informationssicherheit

Modul: 78750 Rechnerorganisation 2

2. Modulkürzel:	-	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	*Rechnerorganisation 1* *Weiteres wird zu Beginn der Veranstaltung festegelegt.*
12. Lernziele:	• *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:	• *Entwurf eines eeinfachen Prozessors mit kommerziellen Entwurfssystemen* • *Umsetzung in Hardware mit einem Field Programmable Gate Array (FPGA) und einem Prototypenboard* • *Entwurf kombinatorischer und sequentieller Schaltungen* • *Programmierung des selbst entworfenen Prozessors in Maschinensprache*
14. Literatur:	*Siehe Veranstaltungshinweise*
15. Lehrveranstaltungen und -formen:	• 787501 Vorlesung Rechnerorganisation 2 • 787502 Übung Rechnerorganisation 2
16. Abschätzung Arbeitsaufwand:	*Präsenszeit: 42 Stunden* *Selbststudium: 138 Stunden* *Gesamt: 180 Stunden*
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), • 78751 Rechnerorganisation 2 (PL), Schriftlich oder Mündlich, Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Parallele Systeme

300 Ergänzungsmodule

Zugeordnete Module:
- 310 Wahlmodule
- 320 Anwendungsfächer
- 330 Wahlmodul aus Master

310 Wahlmodule

Zugeordnete Module:

10030	Architektur von Anwendungssystemen
10060	Computergraphik
10110	Grundlagen der Künstlichen Intelligenz
10170	Imaging Science
10180	Information Retrieval and Text Mining
10210	Mensch-Computer-Interaktion
10220	Modellierung
11620	Automatisierungstechnik I
14390	Programmentwicklung
16500	Software Engineering
17210	Einführung in die Softwaretechnik
36100	Programmierparadigmen
39040	Rechnernetze
40090	Systemkonzepte und -programmierung
42410	Grundlagen des Wissenschaftlichen Rechnens
78640	Grundlagen der Informationssicherheit
78750	Rechnerorganisation 2

Modul: 10030 Architektur von Anwendungssystemen

2. Modulkürzel:	052010002	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann
9. Dozenten:	Frank Leymann
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	

17. Prüfungsnummer/n und -name:	10031 Architektur von Anwendungssystemen (PL), Schriftlich, 90 Min., Gewichtung: 1 [10031] Architektur von Anwendungssystemen (PL), schriftliche Prüfung, 60 Min., Gewicht: 1.0
18. Grundlage für ... :	- Modul Loose Coupling and Message Based Applications - Modul Service Computing - Modul Business Process Management
19. Medienform:	Vorlesungen mit begleitenden Übungen
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 10060 Computergraphik

2. Modulkürzel:	051900002	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Thomas Ertl
9. Dozenten:	Thomas Ertl Daniel Weiskopf Michael Krone Guido Reina
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 Verdeckungsberechung • 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:	• 100602 Übung Computergraphik • 100601 Vorlesung Computergraphik
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 10061 Computergraphik (PL), Schriftlich, 60 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich Prüfungsvorleistung: Übungsschein.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Praktische Informatik (Dialogsysteme)

Modul: 10110 Grundlagen der Künstlichen Intelligenz

2. Modulkürzel:	051900205	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Marc Toussaint
9. Dozenten:	Daniel Hennes Marc Toussaint Andrés Bruhn
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	- Modul 10190 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 • Entscheidungstheorie
14. Literatur:	• S. Russell, P. Norvig, Künstliche Intelligenz: Ein Moderner Ansatz, 3. Aufl., 2012 • S. Russell, P. Norvig,Artificial Intelligence: A Modern Approach, 3rd Edition, 2009
15. Lehrveranstaltungen und -formen:	• 101101 Vorlesung Grundlagen der Künstlichen Intelligenz • 101102 Übung Grundlagen der Künstlichen Intelligenz
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 10111 Grundlagen der Künstlichen Intelligenz (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [10111] Grundlagen der Künstlichen Intelligenz (PL), schriftliche Prüfung, 90 Min., Gewicht: 1.0 Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Autonome Systeme

Modul: 10170 Imaging Science

2. Modulkürzel:	051900210	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Andrés Bruhn
9. Dozenten:	Andrés Bruhn
10. Zuordnung zum Curriculum in diesem Studiengang:	
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.
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
14. Literatur:	• Bässmann, Henning, Kreyss, 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:	• 101702 Übung Imaging Science • 101701 Vorlesung Imaging Science
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 10171 Imaging Science (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [10171] Imaging Science (PL), schriftlich, eventuell mündlich, 120 Min., Gewicht: 1.0,

	Prüfungsvorleistung (USL-V): Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben, schriftlich, eventuell mündlich
18. Grundlage für ... :	Modul Computer VisionModul Correspondence Problems in Computer Vision
19. Medienform:	
20. Angeboten von:	Intelligente Systeme

Modul: 10180 Information Retrieval and Text Mining

2. Modulkürzel:	052401010	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Dr. Roman Klinger
9. Dozenten:	Sebastian Pado Roman Klinger
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	• 10181 Information Retrieval and Text Mining (PL), Schriftlich, Gewichtung: 1 • 10182 Information Retrieval und Text Mining - Hausübungen (USL), Sonstige, Gewichtung: 1 [10181] Information Retrieval und Text Mining (PL), schriftliche Prüfung, 60 Min., Gewicht: 1.0 [10182] Information Retrieval und Text Mining - Hausübungen (USL), Sonstiges
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Theoretische Computerlinguistik

Modul: 10210 Mensch-Computer-Interaktion

2. Modulkürzel:	051900001	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Jun.-Prof. Dr. Niels Henze
9. Dozenten:	Niels Henze Tonja Machulla
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 Dachselt. Interaktive Systeme 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung. Springer, Berlin, 2. Auflage. 2010 • Alan Dix, Janet Finley, Gregory Abowd, Russell Beale, Human-Computer Interaction, 2004 • Ben Shneiderman, Catherine Plaisant, Designing the User Interfaces, 2005
15. Lehrveranstaltungen und -formen:	• 102101 Vorlesung Mensch-Computer-Interaktion • 102102 Übung Mensch-Computer-Interaktion
16. Abschätzung Arbeitsaufwand:	

17. Prüfungsnummer/n und -name:	• 10211 Mensch-Computer-Interaktion (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Soziokognitive Systeme

Modul: 10220 Modellierung

2. Modulkürzel:	052010001	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann
9. Dozenten:	Bernhard Mitschang Frank Leymann Uwe Breitenbücher
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Modul 10280 Programmierung und Software-Entwicklung • Modul 12060 Datenstrukturen und Algorithmen • Modul 40090 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 und komplexe Objekte • Relationenmodell und Relationenalgebra , Überblick SQL - Transformationen von ER nach Relationen, Normalisierung • XML, DTD, XML-Schema, Info-Set, Namensräume • Metamodelle und Repository - RDF, RDF-S und 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 und Design, 2nd Edition, 1994. • H.J. Habermann, F. Leymann, Repository , Oldenbourg 1993. • W. Reisig, Petri-Netze , Vieweg und Teubner 2010. • B. Silver, BPMN Method und Style ,Cody-Cassidy Press 2009.
15. Lehrveranstaltungen und -formen:	• 102202 Übung Modellierung • 102201 Vorlesung Modellierung
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 10221 Modellierung (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [10221] Modellierung (PL), schriftliche Prüfung, 90 Min., Gewicht: 1.0, Prüfungsvorleistung: Übungsschein [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich

18. Grundlage für ... :	Architektur von Anwendungssystemen Datenbanken und Informationssysteme
19. Medienform:	
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 11620 Automatisierungstechnik I

2. Modulkürzel:	050501003	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Michael Weyrich
9. Dozenten:	Michael Weyrich
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Grundlagen der Elektrotechnik, Informatik und Mathematik
12. Lernziele:	Die Studierenden • besitzen grundlegende Kenntnisse über rechnerbasierte Automatisierungssysteme • setzen sich mit Kommunikationssystemen der Automatisierungstechnik ausseinander • wenden grundlegende Methoden und Verfahren der Echtzeit-Programmierung an • lernen spezifische Programmiersprachen der Automatisierungstechnik kennen
13. Inhalt:	• Grundlegende Begriffe der Automatisierungstechnik • Automatisierungs-Gerätesysteme und -strukturen • Prozessperipherie – Schnittstellen zwischen dem Automatisierungscomputersystem und dem technischen Prozess (Prozesssignalerfassung und -überwachung) • Grundlagen zu Kommunikationssystemen in der Automatisierungstechnik (Feldbussysteme, drahtlose Kommunikation) • Grundlagen der Echtzeitprogrammierung (Synchrone und Asynchrone Programmierung, Scheduling-Algorithmen, Synchronisationskonzepte) • Programmiersprachen für die Automatisierungstechnik (Programmierung von Speicherprogrammierbaren und Pneumatischen Steuerungen)
14. Literatur:	• Vorlesungsskript • Lauber, Göhner: Prozessautomatisierung Band 1 (3. Auflage), Springer, 1999 • Früh, Maier: Handbuch der Prozessautomatisierung (3. Auflage) Oldenbourg Industrieverlag, 2004 • Wellenreuther Automatisieren mit SPS (3. Auflage), Vieweg, 2005 • Materialien und Vorlesungsaufzeichnungen im ILIAS
15. Lehrveranstaltungen und -formen:	• 116201 Vorlesung Automatisierungstechnik I • 116202 Übung Automatisierungstechnik I
16. Abschätzung Arbeitsaufwand:	**Präsenzzeit:** 56 h **Selbststudium:** 124 h **Gesamt:** 180 h
17. Prüfungsnummer/n und -name:	11621 Automatisierungstechnik I (PL), Schriftlich, 120 Min., Gewichtung: 1

18. Grundlage für ... :	Automatisierungstechnik II
19. Medienform:	Beamerpräsentation mit Aufzeichnung der Vorlesungen und Übungen
20. Angeboten von:	Automatisierungstechnik und Softwaresysteme

Modul: 14390 Programmentwicklung

2. Modulkürzel:	051520120	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Wagner
9. Dozenten:	Stefan Wagner Jan-Peter Ostberg
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Programmierung und Softwareentwicklung • Einführung in die Softwaretechnik
12. Lernziele:	Die Teilnehmer kennen die Prinzipien der objektorientierten Programmierung und sind in der Lage, Programme in UML zu beschreiben und in Java zu implementieren.
13. Inhalt:	• Grundlagen der objektorientierten Programmierung • Spezifikation und Entwurf objektorientierter Programme mit UML • Vertiefte Programmierung in Java
14. Literatur:	• Rumbaugh, Jacobson, Booch, The unified modeling language reference manual, 2nd ed., 2004 • Rupp, Queins, Zengler, UML 2 glasklar: Praxiswissen für die UML-Modellierung, 3. Aufl. 2007 • Ullenboom: Java ist auch eine Insel. Galileo Computing, 8. Aufl. 2009
15. Lehrveranstaltungen und -formen:	• 143901 Vorlesung Programmentwicklung • 143902 Übung Programmentwicklung
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	14391 Programmentwicklung (PL), Schriftlich, 60 Min., Gewichtung: 1 Klausur 60 min, keine Vorleistungen.
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: 16500 Software Engineering

2. Modulkürzel:	051520110	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Steffen Becker
9. Dozenten:	Steffen Becker André van Hoorn
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Einführung in die Softwaretechnik • Programmierung und Softwarentwicklung
12. Lernziele:	Die Teilnehmer haben tiefe und umfassende Kenntnisse auf dem Gebiet des Software Engineerings.
13. Inhalt:	Ergänzend zur Einführung in die Softwaretechnik und daran anknüpfend, behandelt diese Lehrveranstaltung folgende Themen: • Organisationsaspekte des Software Engineering • Softwareentwicklungsprozesse, Prozessbewertung und -verbesserung • Anforderungsanalyse • Softwarearchitektur • Realisierung und Debugging • Softwarequalitätssicherung • Softwarewartung • Model-Driven Software Development • Weitere ausgewählte Kapitel des Software Engineerings
14. Literatur:	• Summerville, Software Engineering, AW • Ludewig J., Lichter, H., Software Engineering - Grundlagen, Menschen, Prozesse, Techniken, 2. Aufl. 2010 • Liggesmeyer P., Software-Qualität. Testen, Analysieren und Verifizieren von Software. Spektrum Akademischer Verlag, 2002.
15. Lehrveranstaltungen und -formen:	• 165002 Übung Software Engineering • 165001 Vorlesung Software Engineering
16. Abschätzung Arbeitsaufwand:	Vorleszungen und Zentralübungen
17. Prüfungsnummer/n und -name:	16501 Software Engineering (PL), Schriftlich, 90 Min., Gewichtung: 1 90min Klausur
18. Grundlage für ... :	Requirements Engineering and Software ArchitectureModel-Driven Software DevelopmentSoftware Qualität und -Wartung
19. Medienform:	Folien, Videoaufzeichnung, Übungsblätter, elektronische Umfragen und -Tests
20. Angeboten von:	Zuverlässige Software-Systeme

Modul: 17210 Einführung in die Softwaretechnik

2. Modulkürzel:	051520015	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Wagner
9. Dozenten:	Stefan Wagner
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	- Modul 10280 Programmierung und Software-Entwicklung - Modul 12060 Datenstrukturen und Algorithmen - sowie entsprechende Programmiererfahrung
12. Lernziele:	Die Veranstaltung liefert einen ersten Einblick in die Softwaretechnik. Sie ist abgestimmt auf die Software-Qualität im 1. und Programmentwicklung im 3. Semester. Die Teilnehmer kennen die Grundbegriffe der Softwaretechnik und haben wichtige Techniken des Softwareprojekt-Managements und der Software-Entwicklung erlernt. Sie kennen Scrum als eine konkrete Vorgehensweise zur Softwareentwicklung
13. Inhalt:	Die Vorlesung behandelt technische und andere Aspekte der Softwarebearbeitung, wie sie in der Praxis stattfindet. Die einzelnen Themen sind: • Abgrenzung und Motivation des Software Engineerings • Vorgehensmodelle, agiles Vorgehen, Scrum • Software-Management • Software-Prüfung und Qualitätssicherung • Methoden, Sprachen und Werkzeuge für die einzelnen Phasen:Spezifikation, Grobentwurf, Feinentwurf, Implementierung, Test
14. Literatur:	• Ludewig, Lichter: Software Engineering. dpunkt-Verlag, Heidelberg. 2. Aufl. 2010 • Pfleeger, Atlee: Software Engineering. Pearson, 2010 • Rubin: Essential Scrum. Addison-Wesley, 2013
15. Lehrveranstaltungen und -formen:	• 172101 Vorlesung Einführung in die Softwaretechnik • 172102 Übung Einführung in die Softwaretechnik
16. Abschätzung Arbeitsaufwand:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	• 17211 Einführung in die Softwaretechnik (PL), Schriftlich, 60 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich, 30 Min. [17211] Einführung in die Softwaretechnik (PL), schriftliche Prüfung, 60 Min., Gewicht: 1.0, [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich, Hausaufgaben
18. Grundlage für ... :	- Modul Software Engineering - Modul Software-Praktikum
19. Medienform:	• Folien am Beamer unterstützt durch Tafel und Overhead • Dokumente, Links und Diskussionsforum in ILIAS

20. Angeboten von: Software Engineering

Modul: 36100 Programmierparadigmen

2. Modulkürzel:	051510010	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Erhard Plödereder
9. Dozenten:	Erhard Plödereder
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Programmiererfahrung in mindestens einer Programmiersprache, vornehmlich Java, so wie z. B. im Modul "Programmierung und Software Entwicklung"(10280) erworben.
12. Lernziele:	Die Studierenden haben grundlegende Konzepte von Programmiersprachen verstanden, die dem Erlernen weiterer Sprachen und dem vertieften Verständnis ihnen bekannter Sprachen dienlich sind. Sie haben deren Anwendung in mindestens einer weiteren Programmiersprache ihrer Ausbildung verstanden. Sie können ihre Kenntnisse in einfachen Programmen anwenden. Sie können weitere Programmiersprachen in ihrer akademischen und beruflichen Karriere schneller und präziser erlernen.
13. Inhalt:	Überblick typischer Konzepte in Programmiersprachen und ihrer Auswirkungen auf die Sprache und deren Anwendung: • Grundsätzliche Ausführungsmodelle • Speichermodelle und deren Konsequenzen • Datentypen und Typsysteme • unterschiedliche Bindungskonzepte und ihre Auswirkungen • objekt-orientierte Sprachkonzepte • Abstraktion und Kompositionsmechanismen • funktionale Sprachen. • Eventuell werden auch Elemente der parallelen Programmierung und der Logik-Programmierung mit einbezogen. Die Vorlesung ist kein Streifzug durch diverse Programmiersprachen, sondern die Vorstellung zugrundeliegender Prinzipien, und ihrer Begründung aus der Sicht des Software Engineering, insbesondere der Zuverlässigkeit der Anwendung, und, wo nötig, der Implementierungsmodelle.
14. Literatur:	• Sebesta, Robert, Concepts of Programming Languages, Pearson Verlag, 2010 (Hörerschein verfügbar). • weitere Literatur wird zu Beginn der Lehrveranstaltung und auf den Webseiten bekanntgegeben.
15. Lehrveranstaltungen und -formen:	• 361001 Vorlesung Programmierparadigmen • 361002 Übung Programmierparadigmen
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 36101 Programmierparadigmen (PL), Schriftlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich

	[36101] Programmierparadigmen (PL), schriftliche Prüfung, 90 Min., Gewicht: 1.0 [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Programmiersprachen und Übersetzerbau

Modul: 39040 Rechnernetze

2. Modulkürzel:	051200010	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Kurt Rothermel
9. Dozenten:	Kurt Rothermel Frank Dürr
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	- Programmierung und Software-Entwicklung - 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 netzgestützen Systemen anwenden. - Kann sich mit Experten anderer Domänen über Methoden der Rechnernetze verständigen.
13. Inhalt:	• Einführung in die Rechnernetze, ISO Referenzmodell, • Bitübertragungsschicht: Übertragungsmedien, analoge und digitale Informationskodierung und -übertragung, Vermittlungsarten, • Sicherungsschicht: Betriebsarten, Fehlererkennung und -behandlung, Flusskontrolle, • Lokale Netze: CSMA/CD, Token Ring, Token Bus, FDDI, Kopplung, • Vermittlungsschicht: Verbindungsorientierter und verbindungsloser Dienst, Leitwegbestimmung, Überlastkontrolle, • Internetworking, • Internet-Protokoll, • Transportschicht: ausgewählte Realisierungsprobleme und Internet-Protokolle, • Echtzeitkommunikation: IntServ, DiffServ, Sicherheit: Verfahren, IPsec, SSL, TLS.
14. Literatur:	• A.S. Tanenbaum, Computer Networks, 4th Edition, 2003 - D.E. Comer, Computernetzwerke und Internets, 2000 • D.E. Comer, Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture, 1995 • J. F. Kurose, K. W. Ross, Computer Networks: a top-down approach featuring the Internet, 2001 • L.L. Peterson, B.S. Davie, Computer Networks: A Systems Approach, 1999
15. Lehrveranstaltungen und -formen:	• 390401 VL Rechnernetze

	• 390402 ÜB Rechnernetze
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 39041 Rechnernetze (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich Prüfungsdauer: 90 min schriftlich oder 30 min mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Verteilte Systeme

Modul: 40090 Systemkonzepte und -programmierung

2. Modulkürzel:	051200005	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Kurt Rothermel
9. Dozenten:	Kurt Rothermel Frank Dürr
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Modul 10280 Programmierung und Software-Entwicklung • Modul 12060 Datenstrukturen und Algorithmen
12. Lernziele:	• Verstehen grundlegender Architekturen und Organisationsformen von Software-Systemen • Verstehen systemnaher Konzepte und Mechanismen • Kann existierende Systemplattformen und Betriebssysteme hinsichtlich ihrer Eigenschaften analysieren und anwenden. • Kann systemnahe Software entwerfen und implementieren. • Kann nebenläufige Programme entwickeln • Kann mit Experten anderer Fachgebiete die Anwendung von Systemfunktionen abstimmen.
13. Inhalt:	Grundlegende Systemstrukturen - und organisationen • Multitaskingsystem • Multiprozessorsystem • Verteiltes System Modellierung und Analyse nebenläufiger Programme • Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm • Korrektheit- und Leitungskriterien Betriebssystemkonzepte • Organisation von Betriebssystemen • Prozesse und Threads • Eingabe/Ausgabe • Scheduling Konzepte zur Synchronisation über gemeinsamen Speicher • Synchronisationsprobleme und -lösungen • Synchronisationswerkzeuge: Semaphor, Monitor Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer • Taxonomie: Kommunikation und Synchronisation • Nachrichten als Kommunikationskonzept • Höhere Kommunikationskonzepte Basisalgorithmen für Verteilte Systeme • Erkennung globaler Eigenschaften • Schnappschussproblem • Konsistenter globaler Zustand • Verteilte Terminierung Praktische nebenläufige Programmierung in Java • Threads und Synchronisation • Socketschnittstelle • RMI Programmierung
14. Literatur:	Literatur, siehe Webseite zur Veranstaltung

15. Lehrveranstaltungen und -formen:	• 400901 Vorlesung Systemkonzepte und -programmierung • 400902 Übung Systemkonzepte und -programmierung
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 40091 Systemkonzepte und -programmierung (PL), Schriftlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [40091] Systemkonzepte und -programmierung (PL), schriftliche Prüfung, 120 Min., Gewicht: 1.0 [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Verteilte Systeme

Modul: 42410 Grundlagen des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240020	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Dirk Pflüger
9. Dozenten:	Miriam Mehl Stefan Zimmer Dirk Pflüger
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	42411 Grundlagen des Wissenschaftlichen Rechnens (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1 [42411] Grundlagen des Wissenschaftlichen Rechnens (PL), schriftlich oder mündlich, 90 Min., Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Simulation Software Engineering

Modul: 78640 Grundlagen der Informationssicherheit

2. Modulkürzel:	052900001	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. rer. nat. Ralf Küsters
9. Dozenten:	Ralf Küsters
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Empfohlen werden Kenntnisse in den Grundlagen der Informatik und Mathematik wie sie in den ersten beiden Semestern eines Bachelorstudiengangs Informatik/Mathematik vermittelt werden.
12. Lernziele:	Die Veranstaltung soll Studierende einerseits einen Überblick über die Informationssicherheit vermitteln und sie für dieses Thema sensibilisieren. Zum anderen lernen Studierende grundlegende Konzepte der Informationssicherheit kennen.

13. Inhalt:	Die moderne digitale Gesellschaft ist ohne Informationssicherheit nicht denkbar. Daten und sogar Geld sind digital, Kommunikation ist digital, kritische Infrastrukturen (Banken, Industrieanlagen, Verkehrsmittel, etc.) hängen stark von IT-Systemen ab. Es gibt kaum Lebensbereiche, die nicht von der Digitalisierung durchdrungen sind. Die digitale Welt ist deshalb ein attraktives Ziel für Angreifer aller Art (Kriminelle, Geheimdienste, Industriespione, Staaten, etc.) und mittlerweile ständigen Angriffen ausgesetzt. Diese Veranstaltung bietet eine erste Einführung in die Informationssicherheit. Unter anderem werden folgende Themen behandelt, wobei jeweils in der Praxis eingesetze Verfahren betrachtet werden, deren Sicherheit untersucht wird und bekannte Angriffe vorgestellt werden.
	• Grundlagen der Kryptographie
	• (Un-)Sicherheit von Netzwerkprotokollen, wie TCP, DNS, BGP, einschließlich
	  • Denial-of-Service-Angriffe
	  • Firewalls
	• Kryptographische Protokolle (TLS, SSH, WPA2, etc.)
	  • Zertifikate und Public-Key-Infrastrukturen
	  • Authentifizierung und Schlüsselaustausch
	• Zugriffskontrolle, z.B. in Linux, SELinux und Android
	• Grundlagen der Websicherheit
	• Ausblick Systemsicherheit

14. Literatur:	Wird in der Veranstaltung bekanntgegeben.
15. Lehrveranstaltungen und -formen:	• 786401 Vorlesung/Übung zu Grundlagen der Informationssicherheit
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), • 78641 Grundlagen der Informationssicherheit (PL), Schriftlich oder Mündlich, Gewichtung: 1 Unbenotete Studienleistung als Vorleistung (USL-V); ausreichende Punktzahl in den Übungen sowie ggf. in einer Zwischenklausur;

	Prüfungsleistung (PL): Klausur (90 Minuten) oder mündliche Prüfung (30 Minuten) zur Vorlesung und Übung Grundlagen der Informationssicherheit
18. Grundlage für ... :	
19. Medienform:	Projector, blackboard
20. Angeboten von:	Informationssicherheit

Modul: 78750 Rechnerorganisation 2

2. Modulkürzel:	-	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	*Rechnerorganisation 1* *Weiteres wird zu Beginn der Veranstaltung festegelegt.*
12. Lernziele:	• *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:	• *Entwurf eines eeinfachen Prozessors mit kommerziellen Entwurfssystemen* • *Umsetzung in Hardware mit einem Field Programmable Gate Array (FPGA) und einem Prototypenboard* • *Entwurf kombinatorischer und sequentieller Schaltungen* • *Programmierung des selbst entworfenen Prozessors in Maschinensprache*
14. Literatur:	*Siehe Veranstaltungshinweise*
15. Lehrveranstaltungen und -formen:	• 787501 Vorlesung Rechnerorganisation 2 • 787502 Übung Rechnerorganisation 2
16. Abschätzung Arbeitsaufwand:	*Präsenszeit: 42 Stunden* *Selbststudium: 138 Stunden* *Gesamt: 180 Stunden*
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), • 78751 Rechnerorganisation 2 (PL), Schriftlich oder Mündlich, Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Parallele Systeme

320 Anwendungsfächer

Zugeordnete Module: 3201 Anwendungsfach Kraftfahrzeugmechatronik

3201 Anwendungsfach Kraftfahrzeugmechatronik

Zugeordnete Module:
- 13590 Kraftfahrzeuge I + II
- 14130 Kraftfahrzeugmechatronik I + II
- 38370 Grundlagen der Kraftfahrzeugantriebe

Modul: 13590 Kraftfahrzeuge I + II

2. Modulkürzel:	070800001	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Nils Widdecke
9. Dozenten:	Jochen Wiedemann Nils Widdecke
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Kenntnisse aus den Fachsemestern 1 bis 4
12. Lernziele:	Die Studenten kennen die KFZ Grundkomponenten, Fahrwiderstände sowie Fahrgrenzen. Sie können KFZ Grundgleichungen im Kontext anwenden. Die Studenten wissen um die Vor- und Nachteile von Fahrzeug- Antriebs- und Karosseriekonzepte.
13. Inhalt:	Historie des Automobils, Kfz-Entwicklung, Karosserie, Antriebskonzepte, Fahrleistungen - und widerstände, Leistungsangebot, Fahrgrenzen, Räder und Reifen, Bremsen, Kraftübertragung, Fahrwerk, alternative Antriebskonzepte Wichtig: Ab WS2015/16 ist die Prüfung ohne Hilfsmittel zu absolvieren.
14. Literatur:	• Wiedemann, J.: Kraftfahrzeuge I+II, Vorlesungsumdruck, • Braess, H.-H., Seifert, U.: Handbuch Kraftfahrzeugtechnik , Vieweg, 2007 • Bosch: Kraftfahrtechnisches Taschenbuch, 26. Auflage, Vieweg, 2007 • Reimpell, J.: Fahrwerkstechnik: Grundlagen, Vogel-Fachbuchverlag, 2005 • Basshuysen, R. v., Schäfer, F.: Handbuch Verbrennungsmotor, Vieweg, 2007
15. Lehrveranstaltungen und -formen:	• 135901 Vorlesung Kraftfahrzeuge I + II • 135902 Übung Kraftfahrzeuge I + II
16. Abschätzung Arbeitsaufwand:	Vorlesung, Selbststudium
17. Prüfungsnummer/n und -name:	13591 Kraftfahrzeuge I + II (PL), Schriftlich, 120 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	PPT-Präsentation
20. Angeboten von:	Kraftfahrwesen

Modul: 14130 Kraftfahrzeugmechatronik I + II

2. Modulkürzel:	070800002	5. Moduldauer:	Zweisemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Hans-Christian Reuß
9. Dozenten:	Hans-Christian Reuß
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Grundkenntnisse aus den Fachsemestern 1 bis 4
12. Lernziele:	Die Studenten kennen mechatronische Komponenten in Automobilen, können Funktionsweisen und Zusammenhänge erklären. Die Studenten können Entwicklungsmethoden für mechatronische Komponenten im Automobil einordnen und anwenden. Wichtige Entwicklungswerkzeuge können sie nutzen.
13. Inhalt:	**VL Kfz-Mech I:** • kraftfahrzeugspezifische Anforderungen an die Elektronik • Bordnetz (Energiemanagement, Generator, Starter, Batterie, Licht) • Motorelektronik (Zündung, Einspritzung) • Getriebeelektronik • Lenkung • ABS, ASR, ESP, elektromechanische Bremse, Dämpfungsregelung, Reifendrucküberwachung • Sicherheitssysteme (Airbag, Gurt, Alarmanlage, Wegfahrsperre) • Komfortsysteme (Tempomat, Abstandsregelung, Klimaanlage) **VL Kfz-Mech II:** • Grundlagen mechatronischer Systeme (Steuerung/Regelung, diskrete Systeme, Echtzeitsysteme, eingebettete Systeme, vernetzte Systeme) • Systemarchitektur und Fahrzeugentwicklungsprozesse • Kernprozess zur Entwicklung von mechatronischen Systemen und Software (Schwerpunkt V-Modell) **Laborübungen Kraftfahrzeugmechatronik** • Rapid Prototyping (Simulink) • Modellbasierte Funktionsentwicklung mit TargetLink • Elektronik
14. Literatur:	Vorlesungsumdruck: "Kraftfahrzeugmechatronik I" (Reuss) Schäuffele, J., Zurawka, T.: "Automotive Software Engineering" Vieweg, 2006
15. Lehrveranstaltungen und -formen:	• 141302 Vorlesung Kraftfahrzeugmechatronik II • 141301 Vorlesung Kraftfahrzeugmechatronik I • 141303 Laborübungen Kraftfahrzeugmechatronik
16. Abschätzung Arbeitsaufwand:	Vorlesung, Laborübungen, Selbststudium
17. Prüfungsnummer/n und -name:	14131 Kraftfahrzeugmechatronik I + II (PL), Schriftlich, 120 Min., Gewichtung: 1

18. Grundlage für ... :

19. Medienform:	Vorlesung (Beamer), Laborübungen (am PC, betreute Zweiergruppen)
20. Angeboten von:	Kraftfahrzeugmechatronik

Modul: 38370 Grundlagen der Kraftfahrzeugantriebe

2. Modulkürzel:	070810108	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Hubert Fußhoeller
9. Dozenten:	Hubert Fußhoeller
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Keine
12. Lernziele:	Die Studenten kennen Entwicklungen und Design von Otto- und Dieselmotoren vor dem Hintergrund der Gemischbildung, Verbrennung, Schadstoffbildung, etc. Sie können Kennfelder verschiedenster Art interpretieren, Bauteilbelastung und Schadstoffbelastung bzw. deren Vermeidung bestimmen.
13. Inhalt:	Alternative und konventionelle Kraftfahrzeugantriebe, Entwicklungstendenzen (Umweltschutz, Kraftstoffverbrauch). Gemischaufbereitung, Verbrennung, Abgasentgiftung u. Verbrauchsminderung bei Otto- und Dieselmotoren. Schichtladungsmotoren. Kühlung, Schmierung, Motorengeräusch, Nebenaggregate.
14. Literatur:	• Bosch: Kraftfahrtechnisches Taschenbuch, 26. Auflage, Vieweg, 2007 • Basshuysen, R. v., Schäfer, F.: Handbuch Verbrennungsmotor, Vieweg, 2007 • Vorlesungsumdruck
15. Lehrveranstaltungen und -formen:	• 383701 Vorlesung Grundlagen der Kraftfahrzeugantriebe
16. Abschätzung Arbeitsaufwand:	Präsenzzeit56 h, Selbststudium112 h, Gesamt168 h
17. Prüfungsnummer/n und -name:	38371 Grundlagen der Kraftfahrzeugantriebe (PL), Schriftlich, 60 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	Vorlesung (Beamer, Folien, Tafelanschrieb)
20. Angeboten von:	Verbrennungsmotoren und Kraftfahrwesen

330 Wahlmodul aus Master

Zugeordnete Module:

- 10040 Bildsynthese
- 10080 Datenbanken und Informationssysteme
- 10120 Modellbildung und Simulation
- 10250 Parallele Systeme
- 14740 Partielle Differentialgleichungen (Modellierung, Analysis, Simulation)
- 29430 Computer Vision
- 29440 Geometric Modeling and Computer Animation
- 29450 Graphentheorie
- 29460 Algorithmen für die Kryptographie
- 29470 Machine Learning
- 29480 Loose Coupling and Message Based Applications
- 29510 Service Computing
- 29550 Algorithmische Geometrie
- 29570 Computer Interface Technologien
- 29580 Data Compression
- 29590 Digitale Systeme
- 29640 Mikrocontroller
- 29650 Parallele Programmierung
- 29660 Programmanalysen und Compilerbau
- 29670 Rapid Prototyping
- 29680 Real-Time Programming
- 29690 Real-Time Video Processing I
- 29710 Embedded Systems Engineering
- 29720 Mobile Computing
- 29730 Modelling, Simulation, and Specification
- 29740 Fachpraktikum Eingebettete Systeme
- 29750 Fachpraktikum Rechnerarchitektur
- 29760 Algorithmische Gruppentheorie
- 34940 Weiterführende Numerik partieller Differentialgleichungen
- 39250 Distributed Systems I
- 40680 Optimization
- 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
- 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
- 48560 Practical Course Robotics
- 48570 Practical Course Visual Computing
- 48580 Reinforcement Learning
- 48600 Robotics I
- 48620 Scientific Visualization
- 51540 Implementierung Finiter Elemente
- 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
55640 Correspondence Problems in Computer Vision
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 Fachpraktikum Interaktive Systeme
60140 Sprachbau mit Language Workbenches
60860 3D Scanner - Algorithms and Systems
71740 System and Web Security
71760 Security and Privacy
71890 Mikrobiologische Diagnostik in der Humanmedizin
73600 Entwurf Robuster Systeme
73610 Hardwareorientierte Sicherheit
78900 Introduction to Modern Cryptography

Modul: 10040 Bildsynthese

2. Modulkürzel:	051900012	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Thomas Ertl
9. Dozenten:	Thomas Ertl Daniel Weiskopf
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	• 100402 Übung Bildsynthese

	• 100401 Vorlesung Bildsynthese
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 10041 Bildsynthese (PL), Mündlich, 30 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [10041] Bildsynthese (PL), mündliche Prüfung, 30 Min., Gewicht: 1.0 [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich, Übungsschein.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Praktische Informatik (Dialogsysteme)

Modul: 10080 Datenbanken und Informationssysteme

2. Modulkürzel:	051200025	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Bernhard Mitschang
9. Dozenten:	Bernhard Mitschang Holger Schwarz
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Kenntnisse zu Grundlagen der Datenbanken und Informationssysteme beispielsweise aus der Vorlesung "Modellierung" werden vorausgesetzt.
12. Lernziele:	Die Studierenden haben die erforderlichen Kenntnisse für Datenbankprogrammierer in angemessenem Umfang erworben.
13. Inhalt:	Die Vorlesung "Datenbanken und Informationssysteme" ist als Einstiegsveranstaltung in das Vertiefungsgebiet Datenbanksysteme konzipiert. Aufbauend auf dem Inhalt der Vorlesung "Modellierung" werden insbesondere Entwurfs- und Realisierungsaspekte von Datenbanksystemen betrachtet. Die Entwicklung, Installation und Administration von Datenbanksystemen bestimmen hier sowohl Stoffauswahl als auch Detaillierungsgrad. Als Grundlage für alle weiteren Betrachtungen wird ein Schichtenmodell zur Beschreibung eines allgemeinen Datenbanksystems vorgestellt. Darauf aufbauend werden die einzelnen Systemschichten im Detail diskutiert, die dort zu realisierenden Komponenten betrachtet sowie die jeweils vorherrschenden Algorithmen beschrieben und bewertet. Im Einzelnen werden folgende Aspekte vertieft: • Anwendungsprogrammierschnittstelle • Externspeicherverwaltung • DBS-Pufferverwaltung • Speicherungsstrukturen und Zugriffspfadstrukturen • Anfrageverarbeitung und Anfrageoptimierung • Transaktionsverarbeitung, Synchronisation • Logging und Recovery.
14. Literatur:	• A. Kemper, A. Eickler, Datenbanksysteme - Eine Einführung, 2004. • Th. Härder, E. Rahm, Datenbanksysteme, 2008. • H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems. The Complete Book, 2003. • R. Elmasri, S. Navathe, Fundamentals of Database Systems, 2003.
15. Lehrveranstaltungen und -formen:	• 100802 Übung Datenbanken und Informationssysteme • 100801 Vorlesung Datenbanken und Informationssysteme
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 10081 Datenbanken und Informationssysteme (PL), Schriftlich oder Mündlich, 60 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich

	• Schriftliche oder mündliche Prüfungsleistung, 60 Min., Gewicht: 1.0, • Prüfungsvorleistung: Modalitäten werden in der ersten Vorlesung angegeben
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Datenbanken und Informationssysteme

Modul: 10120 Modellbildung und Simulation

2. Modulkürzel:	051240010	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Dirk Pflüger
9. Dozenten:	Miriam Mehl Stefan Zimmer Dirk Pflüger
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	• 101202 Übung Modellbildung und Simulation • 101201 Vorlesung Modellbildung und Simulation
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	10121 Modellbildung und Simulation (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Simulation Software Engineering

Modul: 10250 Parallele Systeme

2. Modulkürzel:	051200065	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch/Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 klassichen Mikroprozessor zur Multi-Core CPUProgrammierung 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:	• 102502 Übung Parallele Systeme • 102501 Vorlesung Parallele Systeme
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	10251 Parallele Systeme (LBP), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Parallele Systeme

Modul: 14740 Partielle Differentialgleichungen (Modellierung, Analysis, Simulation)

2. Modulkürzel:	080300006	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	9 LP	6. Turnus:	Unregelmäßig
4. SWS:	6	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Christian Rohde
9. Dozenten:	Christian Rohde Kunibert Gregor Siebert Bernard Haasdonk Dominik Göddeke
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	*Zulassungsvoraussetzung: Orientierungsprüfung* *Inhaltliche Voraussetzung: Höhere Analysis, Numerische Mathematik 2*
12. Lernziele:	• Grundlagen zur Behandlung von partiellen Differentialgleichungen. • Erwerb von vertieften Fähigkeiten in einem modernen Teilgebiet der Analysis bzw. Numerik, die als Grundlage des Verständnisses aktueller Forschungsthemen dienen.
13. Inhalt:	**Modellierung:** • Herleitung elementarer Typen aus Anwendungen. **Analysis:** • Klassifizierung linearer partieller Differentialgleichungen, elementare Lösungstechniken (Fundamentallösungen, Wellen,...), klassische Existenztheorie in Hölderräumen, schwache Existenztheorie in Sobolevräumen, Asymptotik und qualitatives Verhalten. **Numerik:** • Finite-Differenzen Verfahren, Finite-Elemente Verfahren, effiziente Gleichungslöser. Datenstrukturen,Gittererzeugung.
14. Literatur:	*Wird in der Vorlesung bekannt gegeben.*
15. Lehrveranstaltungen und -formen:	• 147401 Vorlesung Partielle Differentialgleichungen • 147402 Übungen zur Vorlesung Partielle Differentialgleichungen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 63h Selbststudium/Nacharbeitszeit: 187h Prüfungsvorbereitung: 20h **Gesamt: 270h**
17. Prüfungsnummer/n und -name:	14741 Partielle Differentialgleichungen (Modellierung, Analysis, Simulation) (PL), Mündlich, 30 Min., Gewichtung: 1 Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :	
19. Medienform:	

20. Angeboten von: Angewandte Mathematik

Modul: 29430 Computer Vision

2. Modulkürzel:	051900215	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Andrés Bruhn
9. Dozenten:	Andrés Bruhn
10. Zuordnung zum Curriculum in diesem Studiengang:	
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.
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 • Kamerageoemtrie, 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 • Bayessche Entscheidungsthorie der Mustererkennung • Klassifikation mit parametrischen Verfahren, Dichteschätzung • Klassifikation mit nicht-parametrischen Verfahren • Dimensionsreduktion
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:	
17. Prüfungsnummer/n und -name:	• 29431 Computer Vision (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich

	[29431] Computer Vision (PL), schriftlich oder mündlich, 120 Min., Gewicht: 1.0 Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	Correspondence Problems in Computer Vision
19. Medienform:	
20. Angeboten von:	Intelligente Systeme

Modul: 29440 Geometric Modeling and Computer Animation

2. Modulkürzel:	051900010	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch/Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Daniel Weiskopf
9. Dozenten:	Thomas Ertl Daniel Weiskopf Guido Reina
10. Zuordnung zum Curriculum in diesem Studiengang:	
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, sudivision 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 implict 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:	
17. Prüfungsnummer/n und -name:	• 29441 Geometric Modeling and Computer Animation (PL), Mündlich, 30 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [29441] Geometric Modeling and Computer Animation (PL), mündliche Prüfung, 30 Min., Gewicht: 1.0, [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich, Erfolgreiche Teilnahme an Übungen
18. Grundlage für ... :	
19. Medienform:	Video projector, blackboard, exercises using PCs
20. Angeboten von:	Visualisierung

Modul: 29450 Graphentheorie

2. Modulkürzel:	050420105	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	apl. Prof. Dr. rer. nat. habil. Ulrich Hertrampf
9. Dozenten:	Volker Diekert Ulrich Hertrampf Manfred Kufleitner
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), Schriftlich oder Mündlich • 29451 Graphentheorie (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 [29451] Graphentheorie (PL), schriftlich oder mündlich, 120 Min, Gewicht: 1.0 [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich oder mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Theoretische Informatik

Modul: 29460 Algorithmen für die Kryptographie

2. Modulkürzel:	050420110	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	apl. Prof. Dr. rer. nat. habil. Ulrich Hertrampf
9. Dozenten:	Manfred Kufleitner
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Theorie-Vorlesungen des Bachelor-Studiums
12. Lernziele:	Die Studierenden kennen die wichtigsten zahlentheoretischen Algorithmen aus dem Bereich der Kryptographie. Sie können dadurch moderne Verschlüsselungsverfahren anwenden, ihre Sicherheit beurteilen und die Effizienz einstufen.
13. Inhalt:	Die Sicherheit moderner kryptographischer Verfahren basiert in den meisten Fällen auf der Schwierigkeit zahlentheoretischer Probleme. Die Vorlesung behandelt die wichtigsten zahlentheoretischen Algorithmen, und es wird deren Relevanz für die Kryptographie dargestellt. Die Kernthemen sind Primzahltests, Faktorisierung, Wurzelziehen in endlichen Körpern und die Berechnung des diskreten Logarithmus. Zudem werden elliptische Kurven und ihre wichtigsten Eigenschaften vorgestellt. Diese Veranstaltung ergänzt sich gut mit dem Modul "Moderne Kryptographie"; man kann jede der beiden Vorlesungen als erstes hören.
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 Kryprographie, 1999
15. Lehrveranstaltungen und -formen:	• 294601 Vorlesung mit Übungen Algorithmen für die Kryptographie
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 29461 Algorithmen für die Kryptographie (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Theoretische Informatik

Modul: 29470 Machine Learning

2. Modulkürzel:	051200112	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Marc Toussaint
9. Dozenten:	Marc Toussaint
10. Zuordnung zum Curriculum in diesem Studiengang:	
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
14. Literatur:	• *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* 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) • *Pattern Recognition and Machine Learning* 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:	
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), Schriftlich oder Mündlich • 29471 Machine Learning (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Autonome Systeme

Modul: 29480 Loose Coupling and Message Based Applications

2. Modulkürzel:	052010009	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann
9. Dozenten:	Frank Leymann
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 messing 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 patters will be presented, the composability of these patters will 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 Messagin Service API Tutorial und Reference. Addison-Wesley 2001.
15. Lehrveranstaltungen und -formen:	• 294801 Vorlesung mit Übungen Lose Kopplung & Message-basierte Integration
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	29481 Loose Coupling and Message Based Applications (PL), Schriftlich oder Mündlich, 60 Min., Gewichtung: 1 Prüfung schriftlich (60 min) oder mündlich
18. Grundlage für ... :	

19. Medienform:	Lecture and accompanying exercises
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 29510 Service Computing

2. Modulkürzel:	052010010	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann
9. Dozenten:	Frank Leymann
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 und 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:	
17. Prüfungsnummer/n und -name:	29511 Service Computing (PL), Schriftlich oder Mündlich, 60 Min., Gewichtung: 1 schriftlich (60 min) oder mündlich (20 min)
18. Grundlage für ... :	Ausgewählte Themen des Service Computing
19. Medienform:	Lecture and accompanying exercises
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 29550 Algorithmische Geometrie

2. Modulkürzel:	050410105	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Funke
9. Dozenten:	Stefan Funke
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	- Computational Geometry-Algorithms and Applications de Berg, M., Cheong, O., van Kreveld, M., Overmars, M., Springer
15. Lehrveranstaltungen und -formen:	• 295501 Vorlesung Algorithmische Geometrie
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	29551 Algorithmische Geometrie (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 [29551] Algorithmische Geometrie (PL), schriftlich oder mündlich, 120 Min. Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Algorithmik

Modul: 29570 Computer Interface Technologien

2. Modulkürzel:	051230105	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch/Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	29571 Computer Interface Technologien (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1 [29571] Computer Interface Technologien (PL), schriftlich oder mündlich, 90 Min., Gewicht: 1.0, Schriftliche Prüfung von 120 Minuten oder mündliche Prüfung von 30 Min.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Parallele Systeme

Modul: 29580 Data Compression

2. Modulkürzel:	051230110	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	• 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:	
17. Prüfungsnummer/n und -name:	29581 Data Compression (PL), Schriftlich, 90 Min., Gewichtung: 1 [29581] Data Compression (PL), schriftliche Prüfung,90 Min., Gewicht: 1.0 , written 90 Min. or oral 30 Min.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Parallele Systeme

Modul: 29590 Digitale Systeme

2. Modulkürzel:	051230120	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch/Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 Boad und die Realisierung von digitaler Komponenten mittels FPGAs.
13. Inhalt:	• Praktische Einführung in den System-Entwurf mit digitalen Komponenten wie Schnittstellenbausteinen zur Kommunikation, FPGAs, Prozessoren, intelligenten Sensoren etc. • Einführung und Verwendung der Hardware-Beschreibungssprache VHDL zum Entwurf Digitaler Systeme • Digitale Systeme und Board-Integration von digitalen Komponenten • Aufbau von Computer-Boards u. Gbit/s-Interconnects • Entwurf auf höheren Abstraktionsebenen zur schnellen Entwicklung von Prototypen
14. Literatur:	• Kou-Chuan Chang, K.C. Chang, Digital Systems Design with VHDL and Synthesis: An Integrated Approach, 1999 More literature is named in the lecture.
15. Lehrveranstaltungen und -formen:	• 295901 Vorlesung mit Übung Digital System Design I
16. Abschätzung Arbeitsaufwand:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	• 29591 Digitale Systeme (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [29591] Digitale Systeme (PL), schriftlich oder mündlich, 90 Min., Gewicht: 1.0, Schriftliche Prüfung von 120 Min. oder mündliche Prüfung von 30 Min. [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Parallele Systeme

Modul: 29640 Mikrocontroller

2. Modulkürzel:	051230115	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 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 Mikrokontrollern und kennen klassische Architekturen. • Historische Übersicht • Mikrocontroller-Architekturen • Einsatzgebiete von Mikrocontrollern • Befehlssatz klassischer Microcontroller • Assembler-Programmierung von Mikrocontrollern • C-Programmierung von Mikrocontrollern
13. Inhalt:	Microcontrollers (also called micro,Controller, micro,C, MCU) are IC's that combine at least peripheral functions on a sinlge 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 micro,Controller, micro,C, MCU) werden ICs bezeichnet, die mit dem Prozessor mindestens Peripheriefunktionen auf einem Chip vereinen. In vielen Fällen befindet sich der Arbeits- und Programmierspeicher ebenfalls teilweise oder komplett auf dem gleichen Chip. Ein Mikrocontroller ist praktisch ein Ein-Chip-Computersystem. Die Anzahl der verbauten Mikrocontroller überschreitet bei weitem die Zahl der Mikropozessoren. Der Mikrocontroller tritt in Gestalt von eingebetteten Systemen im Alltag oft unbemerkt in technischen Gebrauchsartikeln auf, zum Beispiel in Waschmaschinen, Chipkarten (Geld-, Telefonkarten), Unterhaltungselektronik (Videorekordern, CD-/DVD-Playern, Radios, Fernsehgeräten, Fernbedienungen), Büroelektronik, Kraftfahrzeugen (Steuergeräte für z.B. ABS, Airbag, Motor, Kombiinstrument, ESP usw.), Mobiltelefonen und sogar in Uhren und Armbanduhren. Darüber hinaus sind sie in praktisch allen Computer-Peripheriegeräten enthalten (Tastatur, Maus, Drucker, Monitor, Scanner uvm.). Mikrocontroller sind in Leistung und Ausstattung auf die jeweilige Anwendung angepasst. Daher haben sie gegenüber normalen Computern deutliche Vorteile bei den Kosten und der Leistungsaufnahme. Kleine Mikrocontroller sind in höheren Stückzahlen für deutlich unter 1a,, - 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 Schriftliche Prüfung von 120 Min. oder mündlichen Prüfung von 30 Min.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Parallele Systeme

Modul: 29650 Parallele Programmierung

2. Modulkürzel:	051230130	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Weitere Sprachen

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum in diesem Studiengang:	
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
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Parallele Systeme

Modul: 29660 Programmanalysen und Compilerbau

2. Modulkürzel:	051510311	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Erhard Plödereder
9. Dozenten:	Erhard Plödereder Felix Krause
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 jeweilen 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:	
17. Prüfungsnummer/n und -name:	29661 Programmanalysen und Compilerbau (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Programmiersprachen und Übersetzerbau

Modul: 29670 Rapid Prototyping

2. Modulkürzel:	051230135	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch/Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	
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:	
17. Prüfungsnummer/n und -name:	29671 Rapid Prototyping (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Parallele Systeme

Modul: 29680 Real-Time Programming

2. Modulkürzel:	051510301	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Erhard Plödereder
9. Dozenten:	Erhard Plödereder Felix Krause
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	
17. Prüfungsnummer/n und -name:	29681 Real-Time Programming (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Programmiersprachen und Übersetzerbau

Modul: 29690 Real-Time Video Processing I

2. Modulkürzel:	051230140	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch/Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Sven Simon
9. Dozenten:	Sven Simon
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	This course requires knowledge and experience in (at least) one programming language as well as knowledge of the subject of 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:	
17. Prüfungsnummer/n und -name:	29691 Real-Time Video Processing I (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Parallele Systeme

Modul: 29710 Embedded Systems Engineering

2. Modulkürzel:	051711027	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester/ Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Martin Radetzki
9. Dozenten:	Martin Radetzki
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	
17. Prüfungsnummer/n und -name:	• 29711 Embedded Systems Engineering (Klausur) (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [29711] Embedded Systems Engineering (Klausur) (PL), schriftlich, eventuell mündlich, 120 Min., Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)

Modul: 29720 Mobile Computing

2. Modulkürzel:	051200166	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Kurt Rothermel
9. Dozenten:	Frank Dürr Kurt Rothermel
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	
17. Prüfungsnummer/n und -name:	• 29721 Mobile Computing (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Prüfungsdauer: 90 min schriftlich oder 30 min mündlich
18. Grundlage für ... :	
19. Medienform:	Folien, Tafel
20. Angeboten von:	Verteilte Systeme

Modul: 29730 Modelling, Simulation, and Specification

2. Modulkürzel:	051711020	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Martin Radetzki
9. Dozenten:	Martin Radetzki
10. Zuordnung zum Curriculum in diesem Studiengang:	
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, • specification of timing, concurrency, and non-functional aspects, • object-oriented modeling of embedded systems, • event-driven simulation with the example of the SystemC library, • modeling levels with emphasis on transaction level modeling.
14. Literatur:	• Lecture Notes "Modelling, Simulation, and Specification". • Jantsch: Modeling Embedded Systems and SoCs Concurrency andTime in Models of Computation. Morgan Kaufman Publishers, 2004. • Black, D., Donovan,D.: SystemC from the Ground Up. Kluwer AcademicPublishers, 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 • V Vorleistung (USL-V), Schriftlich oder Mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)

Modul: 29740 Fachpraktikum Eingebettete Systeme

2. Modulkürzel:	051711135	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	6	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Martin Radetzki
9. Dozenten:	wiss. MA
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Modul "Embedded Systems Engineering"
12. Lernziele:	Ability to apply the design methodology and commercial design tools for constructing and analyzing embedded hardware / software systems. Practical experience in software programming and debugging, digital circuit design and verification, usage of lab equipment such as logic analyzers.Experience in preparing structured technical documentation of specifications and designs.
13. Inhalt:	This lab course focuses on analysis, design and implementation of embedded hardware/software systems and issues involved in the development of such systems. 1. Embedded software development 2. Usage of drivers for peripheral components 3. Cross-compilation 4. Remote debugging 5. Software performance profiling 6. Design of accelerator hardware digital circuits 7. Digital circuit simulation 8. FPGA implementation (synthesis) of digital circuits 9. Hardware / software interfacing 10. Integrated functional verification of hardware and software
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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	29741 Fachpraktikum Eingebettete Systeme (LBP), Schriftlich oder Mündlich, Gewichtung: 1 [29741] Fachpraktikum Eingebettete Systeme (LBP), schriftlich oder mündlich, Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)

Modul: 29750 Fachpraktikum Rechnerarchitektur

2. Modulkürzel:	051700025	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Hans-Joachim Wunderlich
9. Dozenten:	Hans-Joachim Wunderlich Rafal Baranowski Chang Liu Laura Rodriguez Gomez
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 und 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:	
17. Prüfungsnummer/n und -name:	29751 Fachpraktikum Rechnerarchitektur (LBP), Sonstige, Gewichtung: 1 [29751] Fachpraktikum Rechnerarchitektur (LBP), schriftlich oder mündlich, Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Rechnerarchitektur

Modul: 29760 Algorithmische Gruppentheorie

2. Modulkürzel:	050420115	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	apl. Prof. Dr. rer. nat. habil. Ulrich Hertrampf
9. Dozenten:	Volker Diekert
10. Zuordnung zum Curriculum in diesem Studiengang:	
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, Heidemannm Verlag 2008. • Lyndon, Schupp: Combinatorial Group Theory, Springer, 1977. • Magnus, Karrass, Solitar: Combinatorial Group Theory, Wiley und 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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h

17. Prüfungsnummer/n und -name:	29761 Algorithmische Gruppentheorie (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 [29761] Algorithmische Gruppentheorie (PL), schriftlich oder mündlich, 120 Min., Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Theoretische Informatik

Modul: 34940 Weiterführende Numerik partieller Differentialgleichungen

2. Modulkürzel:	080803802	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	9 LP	6. Turnus:	Unregelmäßig
4. SWS:	6	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Kunibert Gregor Siebert
9. Dozenten:	Christian Rohde Kunibert Gregor Siebert Bernard Haasdonk
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	empfohlen: Einführung in die Numerik partieller Differentialgleichungen
12. Lernziele:	Die Studenten verfügen über Kenntnis weiterführender Konzepte, Algorithmen und Methoden zur Lösung von partiellen Differentialgleichungen, sie erwerben die Fähigkeit, mit den erlernten Kenntnissen selbständig Methoden zu entwickeln, zu analysieren und umzusetzen, mit denen anwendungsorientierte Probleme effizient und genau gelöst werden
13. Inhalt:	Vertiefende Themen der Numerik für PDEs, beispielsweise aus dem Bereich der Spektralmethoden, Finite Volumen, Continuous und Discontinuous Galerkin, schnelle Löser für dünnbesetzte Systeme, Mehrgitter und Multilevelverfahren, Anwendungen in der Kontinuumsmechanik, hierarchische Ansätze
14. Literatur:	D. Braess, Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie. D. Kröner, Numerical Schemes for Conservation Laws.
15. Lehrveranstaltungen und -formen:	• 349402 Übung Weiterführende Numerik partieller Differentialgleichungen • 349401 Vorlesung Weiterführende Numerik partieller Differentialgleichungen
16. Abschätzung Arbeitsaufwand:	Insgesamt 270 h, wie folgt: Präsenzzeit: 42 h (V), 21 h (Ü) Selbststudium: 207
17. Prüfungsnummer/n und -name:	• 34941 Weiterführende Numerik partieller Differentialgleichungen (PL), Mündlich, 30 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Angewandte Mathematik/Numerik für Höchstleistungsrechner

Modul: 39250 Distributed Systems I

2. Modulkürzel:	051200015	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Kurt Rothermel
9. Dozenten:	Kurt Rothermel Frank Dürr
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	- Programmierung und Software-Entwicklung - Datenstrukturen und Algorithmen - Systemkonzepte und -programmierung
12. Lernziele:	The Students will gain an understanding of the basic charasteristics, 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, synchonization 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) Mulitcast-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:	
17. Prüfungsnummer/n und -name:	• 39251 Distributed Systems I (PL), Schriftlich oder Mündlich, 60 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich

	[39251] Distributed Systems I (PL), schriftlich oder mündlich, 120 Min., Gewicht: 1.0, [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Verteilte Systeme

Modul: 40680 Optimization

2. Modulkürzel:	051200113	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Marc Toussaint
9. Dozenten:	Marc Toussaint
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Solid basic knowledge in linear algebra and analysis. Basic programming skills.
12. Lernziele:	Students will learn to identify, mathematically formalize, and derive algorithmic solutions to optimization problems as they occur in nearly all disciplines, e.g. Machine Learning, Combinatorial Optimization, Computer Vision, Robotics, Simulation. The focus will be on continuous optimization problems (including as they arise from relaxations of discrete problems), including convex problems, quadratic und linear programming, but also non-linear black-box problems. The goal is to give an overview of the various approaches and mathematical formulations and practical experience with the basic paradigms.
13. Inhalt:	Optimization is one of the most fundamental tools of modern sciences. Many phenomena -- be it in computer science, artificial intelligence, logistics, physics, finance, or even psychology and neuroscience -- are typically described in terms of optimality principles. The reason is that it is often easier to describe or design an optimality principle or cost function rather than the system itself. However, if systems are described in terms of optimality principles, the computational problem of optimization becomes central to all these sciences. This lecture aims give an overview and introdution to various approaches to optimization together with practical experience in the exercises. The focus will be on continuous optimization problems and we will cover methods ranging from standard convex optimization and gradient methods to non-linear black box problems (evolutionary algorithms) and optimal global optimization. Students will learn to identify, mathematically formalize, and derive algorithmic solutions to optimization problems as they occur in nearly all disciplines. A preliminary list of topics is: • gradient methods, log-barrier, conjugate gradients, Rprop • constraints, KKT, primal/dual • Linear Programming, simplex algorithm • (sequential) Quadratic Programming • Markov Chain Monte Carlo methods • 2nd order methods, (Gauss-)Newton, (L)BFGS • blackbox stochastic search, including a discussion of evolutionary algorithms
14. Literatur:	
15. Lehrveranstaltungen und -formen:	• 406801 Vorlesung mit Übungen Optimization

16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	40681 Optimization (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Autonome Systeme

Modul: 42420 High Performance Computing

2. Modulkürzel:	051240040	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Dirk Pflüger
9. Dozenten:	Martin Bernreuther Miriam Mehl Dirk Pflüger
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	
17. Prüfungsnummer/n und -name:	42421 High Performance Computing (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1 [42421] High Performance Computing (PL), schriftlich oder mündlich, 90 Min., Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Simulation Software Engineering

Modul: 42460 Numerische Simulation

2. Modulkürzel:	051240060	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Dirk Pflüger
9. Dozenten:	Miriam Mehl Stefan Zimmer Dirk Pflüger
10. Zuordnung zum Curriculum in diesem Studiengang:	
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, Neunhoeffer: 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:	
17. Prüfungsnummer/n und -name:	42461 Numerische Simulation (LBP), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Simulation Software Engineering

Modul: 42480 Ausgewählte Kapitel des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240030	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Dirk Pflüger
9. Dozenten:	Miriam Mehl Dirk Pflüger Stefan Zimmer
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	
17. Prüfungsnummer/n und -name:	42481 Ausgewählte Kapitel des Wissenschaftlichen Rechnens (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1 [42481] Ausgewählte Kapitel des Wissenschaftlichen Rechnens (PL), schriftlich oder mündlich, 90 Min., Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Simulation Software Engineering

Modul: 42900 Business Process Management

2. Modulkürzel:	052010011	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann
9. Dozenten:	Frank Leymann
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Grundlagen der Architektur von Anwendungssystemen, Vorlesung mit Übung, 4 SWS
12. Lernziele:	The course has the objective to provide knowledge about the essentional 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 funcitons (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:	
17. Prüfungsnummer/n und -name:	42901 Business Process Management (PL), Schriftlich oder Mündlich, 60 Min., Gewichtung: 1 schriftlich (60 min) oder mündlich (20 min)

18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 42910 Advanced Business Process Management

2. Modulkürzel:	052010012	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann
9. Dozenten:	Frank Leymann
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	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:	• 429101 Vorlesung mit Übungen, Workflow Management 2
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	42911 Advanced Business Process Management (PL), Schriftlich oder Mündlich, 60 Min., Gewichtung: 1 schriftlich (60 min) oder mündlich (20 min)
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 42920 Hardware-Software-Codesign

2. Modulkürzel:	051711110	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Martin Radetzki
9. Dozenten:	Martin Radetzki
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	42921 Hardware-Software-Codesign (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 [42921] Hardware-Software-Codesign (PL), schriftlich oder mündlich, 120 Min., Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)

Modul: 45740 Rechnernetze II

2. Modulkürzel:	051200168	5. Moduldauer:	Zweisemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Kurt Rothermel
9. Dozenten:	Frank Dürr Kurt Rothermel
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 Bezahlsysteme 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 Schindelhauer, 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:	
17. Prüfungsnummer/n und -name:	45741 Rechnernetze II (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1 Prüfungsdauer: 90 min schriftlich oder 30 min mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Verteilte Systeme

Modul: 45750 Fachpraktikum Verteilte Systeme

2. Modulkürzel:	051200111	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Kurt Rothermel
9. Dozenten:	Frank Dürr
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 und 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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	45751 Fachpraktikum Verteilte Systeme (PL), Sonstige, Gewichtung: 1 [45751] Fachpraktikum Verteilte Systeme (PL), Sonstiges, Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Verteilte Systeme

Modul: 45760 Ausgewählte Kapitel der Algorithmentheorie

2. Modulkürzel:	050410115	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch/Englisch

8. Modulverantwortlicher:	apl. Prof. Dr. rer. nat. habil. Ulrich Hertrampf
9. Dozenten:	Volker Diekert Stefan Funke Ulrich Hertrampf
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	45761 Ausgewählte Kapitel der Algorithmentheorie (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Theoretische Informatik

Modul: 46660 Service Management and Cloud Computing, and Evaluation

2. Modulkürzel:	052010013	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann
9. Dozenten:	Kristof Klöckner
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 Excercise Service Management and Cloud Computing, and Evaluation
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 46661 Service Management and Cloud Computing, and Evaluation (PL), Mündlich, 30 Min., Gewichtung: 1 • V Vorleistung (USL-V), Mündlich, 30 Min. Eine Prüfung kann entweder in 46660 ODER 72340 abgelegt werden, nicht in beiden Modulen. Modul nicht in der Vertiefungslinie wählbar!
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 48480 Data Engineering

2. Modulkürzel:	051210011	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Melanie Herschel
9. Dozenten:	Melanie Herschel
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Lecture Modellierung or comparable course
12. Lernziele:	The students obtain an overview the general data engineering process. Selected system-oriented and algorithmic details for each step and component of the data engineering process are covered such that students get detailed knowledge on possible solutions. The discussion enables students to develop data engineering solutions of their own.
13. Inhalt:	Data engineering involves any data processing necessary to prepare data for subsequent use, e.g., for data analysis. 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 quality and 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 meta-data describing the data processing?
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.Fondations 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:	• 484802 Exersice Data Engineering • 484801 Lecture Data Engineering
16. Abschätzung Arbeitsaufwand:	

17. Prüfungsnummer/n und -name:	48481	Data Engineering (PL), Schriftlich oder Mündlich, 60 Min., Gewichtung: 1
18. Grundlage für ... :		
19. Medienform:		
20. Angeboten von:	Datenbanken und Informationssysteme	

Modul: 48500 Image Synthesis

2. Modulkürzel:	051903654	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Thomas Ertl
9. Dozenten:	Thomas Ertl Daniel Weiskopf
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 (especialy by means of radiosity, distriubtion ray tracing and path tracing)
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. • M. Pharr, G. Humphreys, Physically Based Rendering, 2004.
15. Lehrveranstaltungen und -formen:	• 485002 Exercise Image Synthesis • 485001 Lecture Image Synthesis
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 48501 Image Synthesis (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich

	[48501] Image Synthesis (PL), schriftlich oder mündlich, 120 Min., Gewicht: 1.0 [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich oder mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Visual Computing

Modul: 48550 Practical Course Information Systems

2. Modulkürzel:	051200135	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Bernhard Mitschang
9. Dozenten:	Bernhard Mitschang Holger Schwarz
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Grundlegende Kenntnisse zu Datenbanksystemen, Informationssystemen und Programmiersprachen.
12. Lernziele:	Studierende trainieren den praktischen Umgang mit aktuellen Informationssystemen und lernen typische Aufgaben der Informationsverarbeitung mit diesen Systemen zu bewältigen.Diese praktische Erfahrung ermöglicht es den Studierenden die Informationssysteme in verschiedenen Anwendungsbereichen gezielt einzusetzen.
13. Inhalt:	Der Schwerpunkt dieses Kurses liegt auf dem Entwurf und der Entwicklung datenorientierter Anwendungen. Dies umfasst sowohl Kerndatenbanktechnologie als auch Middleware und Web-Technologie.
14. Literatur:	Will be announced at the beginning of the course
15. Lehrveranstaltungen und -formen:	• 485501 Informationssystem-Fachpraktikum
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	48551 Practical Course Information Systems (LBP), Schriftlich oder Mündlich, Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Datenbanken und Informationssysteme

Modul: 48560 Practical Course Robotics

2. Modulkürzel:	051200222	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Marc Toussaint
9. Dozenten:	Marc Toussaint wiss. Mitarbeiter
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Courses: Robotics I, Reinforcement Learning. Fluency in one programming language, preferrably C++
12. Lernziele:	The Students will gain hand-on experience in programming robots for perception, navigation, planning and object manipulation.
13. Inhalt:	This course will translate the methodological foundations taught in the Robotics I and Reinforcement Learning courses into practical experience with real robots. Students will work on various projects which target at robots that navigate, search for objects and manipulate objects in their environment.
14. Literatur:	
15. Lehrveranstaltungen und -formen:	• 485601 Informationssystem-Fachpraktikum
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	48561 Practical Course Robotics (LBP), Schriftlich oder Mündlich, Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Autonome Systeme

Modul: 48570 Practical Course Visual Computing

2. Modulkürzel:	051900111	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Thomas Ertl
9. Dozenten:	Thomas Ertl
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	
17. Prüfungsnummer/n und -name:	48571 Practical Course Visual Computing (LBP), Schriftlich oder Mündlich, Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Praktische Informatik (Dialogsysteme)

Modul: 48580 Reinforcement Learning

2. Modulkürzel:	051200888	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Marc Toussaint
9. Dozenten:	Vien Ngo
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Solid knowledge in linear algebra, probability theory and optimization. Rough knowledge of Artificial Intelligence. Fluency in at least one programming language.
12. Lernziele:	Students will aquire a deep understanding of Reinforcement Learning methods. Reinforcement Learning addresses the problem of learning optimal behavior (strongly related to optimal control) from data. This course will enable students to apply Reinforcement Learning algorithms in simulated domains and real robotic systems.
13. Inhalt:	Reinforcement Learning considers how an agent, interacting with a world, can improve or learn optimal behavior based on own experience or teacher demonstration. This branch of Artificial Intelligence and Machine Learning has become increasingly important as a foundation of robust intelligent systems and robotics. Optimal exploration (behavior that optimizes the agent's infomation gain) is a particularly interesting aspect of Reinforcement Learning. This lecture will introduce to the theory of Reinforcement Learning and then discuss state-of-the-art algorithms in this area. • Markov Decision Processes and Bellman's optimality principle • relations to stochastic optimal control theory • basic model-free RL methods (TD-Learning, Q-learning, etc) • model-based RL methods • theory of optimal exploration (Bayesian RL, R-max) • relational RL • inverse RL, learning from demonstration and instruction • information theoretic formulations of RL • modern policy search methods (and applications in robotics)
14. Literatur:	• (Main background) R. Sutton and A. Barto, Reinforcement Learning, 1998. This book is freely availableonline. • (For robotics application) S .Thrun, W. Burgard, D. Fox, Probabilistic Robotics, 2006. • (Hardcore theory) C. Szepesvari, Algorithms for Reinforcement Learning, 2010. Draft version is freely availableonline. • S. LaValle, Planning Algorithms, 2006. http://planning.cs.uiuc.edu/
15. Lehrveranstaltungen und -formen:	• 485801 Lecture Reinforcement Learning • 485802 Exercise Reinforcement Learning

16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	48581 Reinforcement Learning (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Autonome Systeme

Modul: 48600 Robotics I

2. Modulkürzel:	051200999	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Marc Toussaint
9. Dozenten:	Marc Toussaint Duy Nguyen-Tuong
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Solid knowledge in linear algebra, probability theory and optimization. Fluency in at least one programming language.
12. Lernziele:	Students will acquire the basic methodologies to model, control and navigate robots, including trajectory planning, control of dynamic systems and object manipulation.
13. Inhalt:	The lecture will give an introduction to robotics, focusing on essential theoretical foundations of planning and controlling motion, state estimation and eventually object manipulation. Exercises in simulations and on a real robot are a core element of this lecture to gain practical experience. • motivation and history • (inverse) kinematics • path finding and trajectory optimization • (non-)holonomic systems • mobile robots • sensor processing (vision, range sensors) • simulation of robots and environments • object grasping and manipulation
14. Literatur:	
15. Lehrveranstaltungen und -formen:	• 486001 Lecture Robotics I • 486002 Exercise Robotics I
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	48601 Robotics I (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Autonome Systeme

Modul: 48620 Scientific Visualization

2. Modulkürzel:	051900777	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Thomas Ertl
9. Dozenten:	Thomas Ertl Daniel Weiskopf Steffen Frey
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), Schriftlich oder Mündlich • 48621 Scientific Visualization (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Praktische Informatik (Dialogsysteme)

Modul: 51540 Implementierung Finiter Elemente

2. Modulkürzel:	080803884	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Dr. Claus-Justus Heine
9. Dozenten:	Claus-Justus Heine
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	empfohlen: "Einführung in die Numerik partieller Differentialgleichungen" oder "Partielle Differentialgleichungen (Modellierung, Analysis, Simulation)"
12. Lernziele:	• Umgang mit gebräuchlichen Finite-Elemente ToolboxenPraktische Umsetzung von Finite-Elemente • Methoden am ComputerValidierung der Implementierung anhand der theoretischen • VorhersagenDarstellung und Visualisierung von Simulationsergebnissen
13. Inhalt:	Die Studierenden erwerben die Fähigkeit, die Diskretisierung partieller Differentialgleichungen mit adaptiven Finite-Elemente Verfahren praktisch am Computer umzusetzen. Die Umsetzung am Computer erfolgt im Rahmen einer gebräuchlichen Finite Elemente Toolbox (z.B. DUNE). Teil der praktischen Umsetzung ist die experimentelle Validierung der numerischen Verfahren und die Visualisierung der Simulationsergebnisse. Die numerischen Verfahren bauen auf den theoretischen Kenntnissen auf, die zum Beispiel in einer der beiden empfohlenen vorangehenden Vorlesungen erworben werden können.
14. Literatur:	Schmidt, A. und Siebert, K. G.: Design of adaptive finite element software Springer, 2005, 42, XII. Braess, D.: Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie, Springer Spektrum, 2013, XVI. Brenner, S. C., Scott, L. R.: The mathematical theory of finite element methods, Springer, 2010, XVII. Weitere Titel nach Bekanntgabe in der Vorlesung
15. Lehrveranstaltungen und -formen:	• 515401 Vorlesung und Übung Implementierung Finiter Elemente
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42h Selbststudium/Nacharbeitszeit: 118h Projektvorstellung mit Vorbereitung: 20h **Gesamt: 180h**
17. Prüfungsnummer/n und -name:	51541 Implementierung Finiter Elemente (BSL), Schriftlich, Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Angewandte Mathematik/Numerik für Höchstleistungsrechner

Modul: 51720 IT-Strategy

2. Modulkürzel:	52010014	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann
9. Dozenten:	Sven Lorenz
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	
12. Lernziele:	Die Vorlesung "Strategisches IT Management (IT Strategie)" vermittelt ein Verständnis von Management-Strategien, Konzepten und Theorien. Sie erläutert das Entwicklen von Strategien und die Bewertung von Optionen unter besonderer Berücksichtigung der Rolle der Informationstechnologie im Zeitalter der Digitalen Transformation. Die Studierenden lernen die Bestandteile einer IT Strategie kennen und sind anschließend in der Lage, aus gegebenen Rahmenbedingungen in einem Unternehmen, wie z. B. der Unternehmensstrategie un der bestehenden IT-Landschaft, systematisch eine IT Strategie abzuleiten und weiterzuentwicklen. Dabei wird sowohl auf die ehemalige, projekthafte Entwicklung einer konkreten IT Strategie im Unternehmen eingegangen, als auch auf das strategische IT Management als permanenter Prozess mit den strategischen Aufgaben in der IT-Organistationsentwicklung, dem IT-Sourcing-Management, dem IT-Architektur-Management, dem IT-Qualitätsmanagement, dem IT-Innovationsmanagement sowie dem IT-Risikomanagement.
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:	• Vorlesungsskript • Helmut Krcmar, "Informationsmanagement", Springer, 2010

	• Jürgen Hofmann, Werner Schmitt, "Masterkurs IT-Management", VIEWEG+TEUBNER, 2010 • Brenner, A. Resch, V. Schulz, "Die Zukunft der IT in Unternehmen", FAZ Buch, 2010 • G. Dern, Management von IT-Architekturen, VIEWEG, 2006 • Martin Kütz, "Kennzahlen in der IT", dpunkt-Verlag, 2007
15. Lehrveranstaltungen und -formen:	• 517201 Vorlesung mit Übungen IT-Strategie
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	51721 IT-Strategy (PL), Schriftlich oder Mündlich, Gewichtung: 1 Prüfungsleistung(PL), Schriftlich (90 min) oder Mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 51740 Quantencomputing

2. Modulkürzel:	050420210	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	apl. Prof. Dr. rer. nat. habil. Ulrich Hertrampf
9. Dozenten:	Ulrich Hertrampf
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 und Sohn, 2008 • Jozef Gruska, "Quantum computing", McGraw-Hill, 1999.
15. Lehrveranstaltungen und -formen:	• 517401 Vorlesung mit Übungen Quantencomputing
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	51741 Quantencomputing (PL), Mündlich, 30 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Theoretische Informatik

Modul: 55600 Advanced Information Management

2. Modulkürzel:	051200099	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch/Englisch

8. Modulverantwortlicher:	PD Dr. Holger Schwarz
9. Dozenten:	Holger Schwarz Bernhard Mitschang
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Kenntnisse zu Grundlagen der Datenbanken und Informationssysteme beispielsweise aus der Vorlesung "Modellierung" werden vorausgesetzt.
12. Lernziele:	Die Studierenden lernen aktuelle Konzepte zur Modellierung, Entwicklung, Verwaltung und Betrieb datenbankorientierter Anwendungen. Hierzu gehören Technologien und Standards zur XML-Verarbeitung und deren Integration in Datenbanksysteme sowie Konzepte und Systeme für Content Management und Datenmanagement in der Cloud.
13. Inhalt:	In dieser Veranstaltung werden insbesondere folgende Themen besprochen: • XML und Datenbanktechnologie (XML-Modellierung, XML-Speicherung, XML-Anfragesprachen, XML-Verarbeitung) • NoSQL Datenmanagement (Key value stores, MapReduce, triple stores, document stores, graph stores) • Content Management (Enterprise Content Management, Information Retrieval, Suchtechnologien)
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:	
17. Prüfungsnummer/n und -name:	• 55601 Advanced Information Management (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich, 90 Min. • Schriftliche (90 min) oder mündliche (30 min) Prüfungsleistung • Prüfungsvorleistung: schriftlich, eventuell mündlich. Details werden zu Beginn der Veranstaltung bekanntgegeben.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Datenbanken und Informationssysteme

Modul: 55610 Information Integration

2. Modulkürzel:	051211001	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Melanie Herschel
9. Dozenten:	Melanie Herschel
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	The integration of heterogeneous data sources, i.e., combining data residing in different data sources to obtain a global view of the data relating to relevant entities, represents one of the major challenges in data management. Especially in the Big Data era, techniques for automatic, efficient, effective, and scalable integration is key to solving the issue of variety. The problem has been considered for decades, and this lecture will cover foundations of data integration as well as algorithmic and system aspects. In particular, this course will cover the following topics: • Distribution, autonomy, and heterogeneity as major challenges in data integration. • Types of data integration and associated architectures of integrating systems. • Query processing in integrating systems. • Overcoming schematic heterogeneities between integrated data sources (schema mapping and schema matching). • Getting a unified view of the data using duplicate detection and data fusion.
14. Literatur:	AnHai Doan and Alon Halevy and Zachary Ives. Principles of Data IntegrationMorgan Kaufmann;, 2012, ISBN 0124160441. Ulf Leser, Felix Naumann: Informationsintegration: Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen, dpunkt Verlag, 2006, ISBN 3898644006.
15. Lehrveranstaltungen und -formen:	• 556102 Übung Information Integration • 556101 Vorlesung Information Integration
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	55611 Information Integration (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1 [55611] Information Integration (PL), schriftlich oder mündlich, 90 Min., Gewicht: 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:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Bernhard Mitschang
9. Dozenten:	Bernhard Mitschang Holger Schwarz
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Kenntnisse zu Grundlagen der Datenbanken und Informationssysteme beispielsweise aus der Vorlesung "Modellierung" werden vorausgesetzt.
12. Lernziele:	Die Studierenden verstehen die Herausforderungen, die sich bei der Integration der Daten aus heterogenen Datenquellen in ein konsolidiertes Data Warehouse ergeben. Sie kennen die typische Data-Warehouse-Architektur und aktuelle Trends, wie z.B. Echtzeit-Reporting. Ebenso kennen sie die Struktur eines Data Warehouse und die wichtigsten Prozesse, um ein solches aufzubauen (Extraktion, Transformation, Laden). Die Studierenden haben darüber hinaus einen Überblick über die wichtigsten Technologien, um Daten in einem Data Warehouse zu analysieren. Hierzu gehört Reporting, Online Analytic Processing und Data Mining.
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. Further literature 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:	
17. Prüfungsnummer/n und -name:	• 55621 Data Warehousing, Data Mining, and OLAP (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich, 60 Min. • Schriftliche (90 min) oder mündliche (30 min) Prüfungsleistung

	• Prüfungsvorleistung: schriftlich, eventuell mündlich. Details werden zu Beginn der Veranstaltung bekanntgegeben.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Datenbanken und Informationssysteme

Modul: 55630 Information Visualization and Visual Analytics

2. Modulkürzel:	051900099	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Daniel Weiskopf
9. Dozenten:	Thomas Ertl Daniel Weiskopf Steffen Koch
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 Infomation • Robert Spence. Design for Interaction • Jim Thomas. Illuminating the Path
15. Lehrveranstaltungen und -formen:	• 556301 Vorlesung und Übung Informationsvisualisierung
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 55631 Information Visualization and Visual Analytics (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich Erfolgreiche Übungsteilnahmen / excercises passed
18. Grundlage für ... :	
19. Medienform:	Video projector, blackboard, exercises using PCs
20. Angeboten von:	Visualisierung

Modul: 55640 Correspondence Problems in Computer Vision

2. Modulkürzel:	051900211	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	6	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Andrés Bruhn
9. Dozenten:	Andrés Bruhn
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	• Modul 10190 Mathematik für Informatiker und Softwaretechniker • Modul 10170 Imaging Science - Modul 29430 Computer Vision
12. Lernziele:	Der Student kann Korrespondenzprobleme im Computer-Vision-Bereich selbständig einordnen, Lösungsstrategien mathematisch modellieren und diese dann geeignet algorithmisch umsetzen.
13. Inhalt:	• Basisverfahren: Block Matching, Detektion von Verdeckungen, Merkmalsfindung, Feature Matching • Optischer Fluss: Lokale und Globale differentiale Verfahren, Parametrisierungsmodelle, Konstanzannahmem, Daten- und Glattheitsterme, Numerik, Große Verschiebungen, Hochgenaue Verfahren • Stereorekonstruktion: Projektive Geometrie, Epipolargeometrie, Schätzung der Fundementalmatrix • Szenenfluss: Gemeinsame Schätzung von Struktur, Bewegung und Geometrie • Medizinische Bildregistrierung: Mutual Information, Elastische und krümmungsbasierte Regularisierung, Landmarks • Particle Image Velocimetry: Div-Curl-Regularisierung, Inkompressibler Navier Stokes Prior
14. Literatur:	• O. Faugeras, Q.-T. Luong: The Geometry of Multiple Images, 2001. • J. Modersitzki: Numerical Methods for Image Registration, 2003. • A. Bruhn: Variational Optic Flow Computation: Accurate Modeling and Efficient Numerics, Ph.D. Thesis, 2006.
15. Lehrveranstaltungen und -formen:	• 556401 Vorlesung Correspondence Problems in Computer Vision • 556402 Übung Correspondence Problems in Computer Vision
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 55641 Correspondence Problems in Computer Vision (PL), Schriftlich oder Mündlich, 120 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [55641] Correspondence Problems in Computer Vision (PL), schriftlich, eventuell mündlich, 120 Min., Gewicht: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Intelligente Systeme

Modul: 55650 Multimodal Interaction for Ubiquitous Computers

2. Modulkürzel:	051900033	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Jun.-Prof. Dr. Niels Henze
9. Dozenten:	Niels Henze Pawel Wozniak
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	
17. Prüfungsnummer/n und -name:	55651 Multimodal Interaction for Ubiquitous Computers (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Soziokognitive Systeme

Modul: 55740 Advanced Service Computing

2. Modulkürzel:	052010015	5. Moduldauer:	Zweisemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	5	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann
9. Dozenten:	Frank Leymann
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 und Sons, 3rd Edition, 2010

15. Lehrveranstaltungen und -formen:	• 557401 Advanced Service Computing Lecture (Summer) • 557402 Lecture Services and Security (Winter)
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	• 55741 Advanced Service Computing (PL), Schriftlich oder Mündlich, 60 Min., Gewichtung: 1 • V Vorleistung (USL-V), schriftlich (60 min) oder mündlich (20 min)
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Simulation Workflows

Modul: 56680 Automaten über unendlichen Objekten

2. Modulkürzel:	050420230	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	apl. Prof. Dr. rer. nat. habil. Ulrich Hertrampf
9. Dozenten:	Volker Diekert
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	56681 Automaten über unendlichen Objekten (PL), Mündlich, 30 Min., Gewichtung: 1 [56681] Automaten über unendlichen Objekten (PL), mündliche Prüfung, 30 Min., Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Theoretische Informatik

Modul: 56790 Parallele Numerik

2. Modulkürzel:	051240080	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Miriam Mehl
9. Dozenten:	Miriam Mehl Dirk Pflüger Stefan Zimmer
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	

17. Prüfungsnummer/n und -name:	56791 Parallele Numerik (PL), Schriftlich, 90 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Simulation großer Systeme

Modul: 56980 Fachpraktikum Cloud Architekturen und Management

2. Modulkürzel:	052010016	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Frank Leymann
9. Dozenten:	Johannes Wettinger
10. Zuordnung zum Curriculum in diesem Studiengang:	
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-sentric 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:	
17. Prüfungsnummer/n und -name:	56981 Fachpraktikum Cloud Architekturen und Management (LBP), Sonstige, 0 Min., Gewichtung: 1 Fachpraktikum Cloud Architekturen und Management (LBP), Studienbegleitend, Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 57050 Compilerbau

2. Modulkürzel:	051010201	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Erhard Plödereder
9. Dozenten:	Erhard Plödereder
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	57051 Compilerbau (PL), Schriftlich oder Mündlich, 90 Min., Gewichtung: 1 57051] Compilerbau (PL), schriftliche Prüfung, 90 Min., Gewicht: 1.0
18. Grundlage für ... :	Programmanalysen und Compilerbau
19. Medienform:	
20. Angeboten von:	Programmiersprachen und Übersetzerbau

Modul: 57680 Einführung in die Chaostheorie

2. Modulkürzel:	074810350	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Frank Allgöwer
9. Dozenten:	Viktor Avrutin
10. Zuordnung zum Curriculum in diesem Studiengang:	
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-glatten 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-glatten Systemen), Bifurkations-szenarien (in glatten und stückweise-glatten 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), Mündlich, 30 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Systemtheorie und Regelungstechnik

Modul: 58190 Entwurf und Implementierung eines Compilers

2. Modulkürzel:	05151313	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Unregelmäßig
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Erhard Plödereder
9. Dozenten:	Erhard Plödereder Timm Felden
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Modul Compilerbau ist notwendige Voraussetzung, Java-Kennntnisse 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, Zwischencodegenerierung, 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:	Präsenzstunden: 42 h Eigenstudiumstunden: 138 h Gesamtstunden: 180 h
17. Prüfungsnummer/n und -name:	• 58191 Entwurf und Implementierung eines Compilers (PL), Mündlich, 30 Min., Gewichtung: 1 • V Vorleistung (USL-V), Schriftlich oder Mündlich [58191] Entwurf und Implementierung eines Compilers (PL), mündliche Prüfung, 30 Min. , Gewicht: 1.0 [Prüfungsvorleistung] Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Programmiersprachen und Übersetzerbau

Modul: 58440 Fachpraktikum: Algorithmik

2. Modulkürzel:	050410101	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	6	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Funke
9. Dozenten:	Stefan Funke
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	
12. Lernziele:	
13. Inhalt:	
14. Literatur:	wird in der Veranstaltung bekanntgegeben
15. Lehrveranstaltungen und -formen:	• 584401 Fachpraktikum Algorithmik
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	58441 Fachpraktikum: Algorithmik (LBP), Sonstige, Gewichtung: 1 [58441] Fachpraktikum: Algorithmik (LBP), schriftlich und mündlich, Gewicht: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Algorithmik

Modul: 60120 Fachpraktikum Interaktive Systeme

2. Modulkürzel:	051900010	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester/ Sommersemester
4. SWS:	4	7. Sprache:	Deutsch/Englisch

8. Modulverantwortlicher:	Jun.-Prof. Dr. Niels Henze
9. Dozenten:	Niels Henze wiss. Mitarbeiter
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	
17. Prüfungsnummer/n und -name:	60121 Fachpraktikum Interaktive Systeme (LBP), Sonstige, Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Soziokognitive Systeme

Modul: 60140 Sprachbau mit Language Workbenches

2. Modulkürzel:	051520020	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Wagner
9. Dozenten:	Markus Völter
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Compilerbau Objektorientierte Programmierung
12. Lernziele:	Die Studenten verstehen, warum und wie Modelle für den praktizierenden Softwareentickler 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.
13. Inhalt:	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. Die Veranstaltung wird als Blockveranstaltung als Workshop ausgeführt, viele praktische Anteile. Die Klausur findet direkt am Ende der Blockveranstaltung statt. Ort ist bei der itemis AG, Industriestrasse 6, Vaihingen (direkt neben dem Bhf)
14. Literatur:	Buch http://dslbook.org/ + ggfs. diverse wissenschaftliche Papiere
15. Lehrveranstaltungen und -formen:	• 601401 Vorlesung Sprachbau • 601402 Übung Sprachbau
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	60141 Sprachbau mit Language Workbenches (PL), Schriftlich, 60 Min., Gewichtung: 1
18. Grundlage für ... :	
19. Medienform:	Powerpoint, Tafel, Demos, Diskussionen, Selbstarbeit der Studenten
20. Angeboten von:	Software Engineering

Modul: 60860 3D Scanner - Algorithms and Systems

2. Modulkürzel:	051230002	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr.-Ing. Sven Simon
9. Dozenten:	
10. Zuordnung zum Curriculum in diesem Studiengang:	
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), Schriftlich, 90 Min., Gewichtung: 1 Prüfungsleistung (PL): schriftlich, 90 min.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Parallele Systeme

Modul: 71740 System and Web Security

2. Modulkürzel:	052900002	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. rer. nat. Ralf Küsters
9. Dozenten:	Ralf Küsters
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Solide Kenntnisse in mindestens einer Programmiersprache.
12. Lernziele:	• Students are sensitized for common security vulnerabilities and attack vectors in computer systems and the web, • Students are familar with concrete attacks on computer systems and the web, and understand the underlying principles, • Students are familiar with common defense mechanisms.
13. Inhalt:	IT-systems are constantly under attack, by various kinds of attackers with diverse interests: criminal organizations with monetary interests, intelligence agencies, industrial espionage by states and companies. The course covers the most common attack vectors on computer systems, including mobile devices, and the web, including, for example, stack and heap overflows, format string vulnerabilities, integer overflows, return-oriented-programming, Cross-Site-Scripting (CSS/XSS), SQL Injections, and Cross-Site-Request-Forgery (XSRF), etc. The course also discusses common defense mechanisms, including, for example, access control mechanisms, address space layout randomization (ASLR), static code analysis, security monitoring, input/output sanitization, prepared statements, etc.
14. Literatur:	Will be announced in class
15. Lehrveranstaltungen und -formen:	• 717401 Vorlesung System and Web Security • 717402 Übung System and Web Security
16. Abschätzung Arbeitsaufwand:	Vorlesung und Übung System- und Websicherheit
17. Prüfungsnummer/n und -name:	• 71741 System and Web Security (PL), Schriftlich, Gewichtung: 1 • V Vorleistung (USL-V), Unbenotete Studienleistung als Vorleistung (USL-V); ausreichende Punktzahl in den Übungen Prüfungsleistung (PL): Klausur (90 Minuten) zur Vorlesung und Übung System- und Websicherheit
18. Grundlage für ... :	
19. Medienform:	Projektor, Tafel
20. Angeboten von:	Informationssicherheit

Modul: 71760 Security and Privacy

2. Modulkürzel:	052900004	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Sommersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. rer. nat. Ralf Küsters
9. Dozenten:	Ralf Küsters
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Kenntnisse aus den Vorlesungen *Grundlagen der Informationssicherheit* (Bachelor) sowie *Introduction to Modern Cryptography* (Master) sind vorteilhaft, werden allerdings nicht zwingend vorausgesetzt. Die Veranstaltung verlangt solide Kenntnisse in den Grundlagen der Informatik und der Mathematik wie sie in den ersten vier Semestern eines Bachelorstudiengangs in Informatik (oder Mathematik) vermittelt werden.
12. Lernziele:	Students will acquire an in-depth understanding of central topics in information security and privacy.
13. Inhalt:	This course covers some of the most important, typically advanced topics in information security and privacy. The selection of topics can vary from term to term, depending on the development of the field and the focus of the information security group. Possible topics include: • Zero-Knowledge Protocols: a fundamental concept in many advanced secure and privacy preserving systems • Verification of cryptographic protocols: What does it mean for protocols, such as TLS, to be secure? How can we prove security? Can we prove security using automated tools? • Blockchains, Smart Contracts, and applications, such as cryptocurrencies, e.g., Bitcoin and Ethereum. • Secure Multi-Party Computation: how can multiple parties compute a common function without revealing their input? E.g., how can two millionaires figure out who earns more without revealing their income to each other? • Differential Privacy and Privacy-Preserving Data Mining: how to make use of information in (statistical) databases without revealing information about individuals? • E-Voting: Can we have a system where voters can make sure that their votes were actually counted even when the voting servers are completely malicious? • Web-based security protocols, such as web-based single-sign on protocols • Advanced attacks and defenses in as well as models of web security
14. Literatur:	Will be announced in class.
15. Lehrveranstaltungen und -formen:	• 717601 Vorlesung Security and Privacy • 717602 Übung Security and Privacy
16. Abschätzung Arbeitsaufwand:	Vorlesung und Übung zu Security and Privacy

17. Prüfungsnummer/n und -name:	• 71761 Security and Privacy (PL), Schriftlich oder Mündlich, Gewichtung: 1 • V Vorleistung (USL-V), Unbenotete Studienleistung als Vorleistung (USL-V); ausreichende Punktzahl in den Übungen Prüfungsleistung (PL): Klausur (90 Minuten) oder mündliche Prüfung (30 Minuten) zur Vorlesung und Übung Security and Privacy
18. Grundlage für ... :	
19. Medienform:	Projektor, Tafel
20. Angeboten von:	Informationssicherheit

Modul: 71890 Mikrobiologische Diagnostik in der Humanmedizin

2. Modulkürzel:	-	5. Moduldauer:	-
3. Leistungspunkte:	6 LP	6. Turnus:	-
4. SWS:	0	7. Sprache:	-

8. Modulverantwortlicher:
9. Dozenten:
10. Zuordnung zum Curriculum in diesem Studiengang:
11. Empfohlene Voraussetzungen:
12. Lernziele:
13. Inhalt:
14. Literatur:
15. Lehrveranstaltungen und -formen:
16. Abschätzung Arbeitsaufwand:
17. Prüfungsnummer/n und -name:
18. Grundlage für ... :
19. Medienform:
20. Angeboten von:

Modul: 73600 Entwurf Robuster Systeme

2. Modulkürzel:	20909003	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Jedes 2. Sommersemester
4. SWS:	-	7. Sprache:	Englisch

8. Modulverantwortlicher:	
9. Dozenten:	Prof. Dr. Ilia Polian, Hardwareorientierte Informatik
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Grundkenntnisse der Technischen Informatik
12. Lernziele:	Die Studierenden lernen diverse Ausprägungen von Robustheit, Zuverlässigkeit, Verlässlichkeit und Testbarkeit in komplexen elektronischen Systemen kennen. Sie werden mit relevanten Metriken und Methoden zur qualitativen und quantitativen Bewertung dieser Attribute vertraut gemacht. Sie sind in der Lage, wissenschaftliche Beiträge aus den genannten Bereichen zu verstehen.
13. Inhalt:	- Testmethoden: Fehlermodellierung, Fehlersimulation, automatische Testmustergenerierung, Fehlerdiagnose - Testgerechter Entwurf, eingebauter Selbsttest, Testdatenkompression - Zuverlässigkeitstheorie und Redundanztechniken: Hardware-, Informations-, Zeit- und Software-Redundanz
14. Literatur:	Vorlesungsfolien / Lecture slides (in Englisch) Abramovici/Breuer/Friedman, Digital System Testing and Testable Design Eggersglüß/Fey/Polian, Test digitaler Hardware Koren/Krishna, Fault-tolerant Systems
15. Lehrveranstaltungen und -formen:	• 736001 Entwurf Robuster Systeme, Vorlesung • 736002 Entwurf Robuster Systeme, Übung
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	73601 Entwurf Robuster Systeme (PL), , 90 Min., Gewichtung: 1 Prüfungsleistung (PL): Klausur (90 Minuten) oder mündliche Prüfung (25 Minuten) zur Vorlesung „Entwurf Robuster Systeme“
18. Grundlage für ... :	
19. Medienform:	Powerpoint-Folien, Tafelanschrieb
20. Angeboten von:	

Modul: 73610 Hardwareorientierte Sicherheit

2. Modulkürzel:	20909002	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	-	6. Turnus:	Jedes 2. Sommersemester
4. SWS:	-	7. Sprache:	Englisch

8. Modulverantwortlicher:	Prof. Dr. Ilia Polian
9. Dozenten:	Prof. Dr. Ilia Polian
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Grundkenntnisse der Technischen Informatik; wünschenswert sind Kenntnisse der Kryptologie und der IT-Sicherheit
12. Lernziele:	Die Studierenden lernen den Einsatz von Hardware-Bausteinen zur Erreichung von Sicherheitszielen für elektronische Systeme kennen, darunter kryptografische Blöcke und Lösungen zur sicheren Erzeugung, Daten Isolation kritischer Daten. Außerdem werden Angriffsszenarien diskutiert, bei welchen die Hardwareblöcke eine tragende Rolle spielen, und Gegenmaßnahmen gegen solche Angriffe vorgestellt. Die Teilnehmer sind ferner in der Lage, wissenschaftliche Beiträge aus den genannten Bereichen zu verstehen.
13. Inhalt:	- Hardware-Realisierungen kryptografischer Verfahren - Hardware-Bausteine für sichere Systeme (RNG, PUF) - Sicherheitsorientierte Architekturen - Seitenkanal- und Fehlerinjektionsangriffe, Gegenmaßnahmen - Angriffe auf Wertschöpfungskette, Gegenmaßnahmen
14. Literatur:	Vorlesungsfolien / Lecture slides (in Englisch) Tehranipoor/Wang, Introduction to Hardware Security and Trust
15. Lehrveranstaltungen und -formen:	• 736101 Hardwareorientierte Sicherheit, Vorlesung • 736102 Hardwareorientierte Sicherheit, Übung
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	73611 Hardwareorientierte Sicherheit (PL), , 90 Min., Gewichtung: 1 Prüfungsleistung (PL): Klausur (90 Minuten) oder mündliche Prüfung (25 Minuten) zur Vorlesung „Hardwareorientierte Sicherheit“
18. Grundlage für ... :	
19. Medienform:	Powerpoint-Folien, Tafelanschrieb
20. Angeboten von:	

Modul: 78900 Introduction to Modern Cryptography

2. Modulkürzel:	052900003	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester
4. SWS:	4	7. Sprache:	Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. rer. nat. Ralf Küsters
9. Dozenten:	Ralf Küsters
10. Zuordnung zum Curriculum in diesem Studiengang:	
11. Empfohlene Voraussetzungen:	Die Veranstaltung verlangt solide Kenntnisse in den Grundlagen der Mathematik wie sie in den ersten drei oder vier Semestern eines Bachelorstudiengangs in Informatik/Mathematik vermittelt werden. Kenntnisse der Inhalte der Vorlesung *Grundlagen der Informationssicherheit* sind nützlich, aber keine zwingende Voraussetzung.
12. Lernziele:	Students will acquire an in-depth understanding of cryptography. They will be able to judge and assess the security of cryptographic constructions used in practice (encryption schemes, digital signatures, messages authentication codes, etc.) and will be able to read scientific papers on cryptography.
13. Inhalt:	p { margin-bottom: 0.1in; direction: ltr; color: rgb(0, 0, 10); line-height: 120%; text-align: left; }p.western { font-family: "Calibri", serif; font-size: 11pt; }p.cjk { font-family: "Times New Roman"; font-size: 11pt; }p.ctl { font-family: "Times New Roman"; font-size: 11pt; }a:link { color: rgb(0, 0, 255); }a.ctl:link { font-family: "Times New Roman"; } Cryptography is everywhere! We heavily rely on cryptography in our everyday life when we do, for example, online shopping and online banking, pay with credit or debit card, open doors with electronic keys, or when we use social networks, instant messengers, online games, WiFi, mobile networks, or electronic currencies. Here, cryptography is essential in order to guarantee various central security properties such as secrecy and integrity of messages as well as authenticity of the communication partners. This course provides an introduction to modern cryptography. In the traditional approach to cryptography, cryptographers proposed, for example, encryption algorithms, and then others, cryptanalysts, tried to break them. In modern cryptography, cryptographers try to prove that their cryptographic constructions are secure under certain assumptions, even when attacked by powerful adversaries. Hence, cryptography turned from pure art to science. The course covers several fundamental cryptographic primitives, including (symmetric and asymmetric) encryption, hash functions, digital signatures, and message authentication codes. These primitives are important building blocks for other cryptographic constructions and for cryptographic protocols (TLS, SSH, WPA2, etc.), used by billions of people every day. The course presents common cryptographic constructions as used in practice, such as AES with various encryption modes (e.g., CBC, CTR), RSA,

	ElGamal, HMAC, PKCS#1, DSA. It also discusses public-key infrastructures and cryptographic protocols. In the spirit of modern cryptography, we ask the following questions: What does it mean for an encryption algorithm, digital signature, etc. to be secure? Under which assumptions can we prove security? For several cryptographic constructions used in practice, including those mentioned above, we prove security or present attacks. This provides a deep understanding of the security/insecurity of the cryptography that surrounds us.
14. Literatur:	• Ralf Küsters and Thomas Wilke. Moderne Kryptographie - Eine Einführung. Vieweg + Teubner, 2011. • Jonathan Katz and Yehuda Lindell. Introduction to Modern Cryptography - Second Edition. CRC Press 2015.
15. Lehrveranstaltungen und -formen:	• 789001 Vorlesung und Übung zu Introduction to Modern Cryptography
16. Abschätzung Arbeitsaufwand:	Vorlesung und Übung zu Introduction to Modern Cryptography
17. Prüfungsnummer/n und -name:	• V Vorleistung (USL-V), • 78901 Introduction to Modern Cryptography (PL), Schriftlich oder Mündlich, Gewichtung: 1 Unbenotete Studienleistung als Vorleistung (USL-V); ausreichende Punktzahl in den Übungen Prüfungsleistung (PL): Klausur (90 Minuten) oder mündliche Prüfung (30 Minuten) zur Vorlesung und Übung Introduction to Modern Cryptography
18. Grundlage für ... :	
19. Medienform:	Projector, blackboard
20. Angeboten von:	Informationssicherheit

400 Schlüsselqualifikationen fachaffin

Zugeordnete Module:
- 38610 Seminar-INF 1
- 78610 Bachelor-Forschungsprojekt Informatik

Modul: 38610 Seminar-INF 1

2. Modulkürzel:	050420095	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	3 LP	6. Turnus:	Wintersemester/ Sommersemester
4. SWS:	2	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Funke
9. Dozenten:	Dozenten der Informatik
10. Zuordnung zum Curriculum in diesem Studiengang:	
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:	• 386101 Seminar
16. Abschätzung Arbeitsaufwand:	
17. Prüfungsnummer/n und -name:	38611 Seminar-INF 1 (BSL), Sonstige, Gewichtung: 1 Präsentation im Seminar und Abgabe einer Ausarbeitung am Semesterende, Gewichtung: 1.0
18. Grundlage für ... :	

19. Medienform:

20. Angeboten von:	Formale Methoden der Informatik

Modul: 78610 Bachelor-Forschungsprojekt Informatik

2. Modulkürzel:	051900012	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	6 LP	6. Turnus:	Wintersemester/ Sommersemester
4. SWS:	4	7. Sprache:	Deutsch

8. Modulverantwortlicher:	Jun.-Prof. Dr. Niels Henze
9. Dozenten:	Dozenten der Informatik
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 oder einer Frage aus der Praxis 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 festgehalten werden. 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:	• Leedy, Ormrod. Practical Research: Planning and Design. Pearson Prentice Hall, 2009. • Marcus Deininger, Horst Lichter, Jochen Ludewig, Kurt Schneider. Studienarbeiten. vdf Hochschulverlag, 2005.
15. Lehrveranstaltungen und -formen:	• 786101 Bachelor-Forschungsprojekt Informatik, Praktikum

16. Abschätzung Arbeitsaufwand:	Projektarbeit im kleinen Team
17. Prüfungsnummer/n und -name:	78611 Bachelor-Forschungsprojekt Informatik (PL), Sonstige, Gewichtung: 1 Aktive Mitwirkung im Projektteam. Abgabe eines Projektberichts in Form einer wissenschaftlichen Publikation. Teilnahme und Mitwirkung an der internen Semesterkonferenz.
18. Grundlage für ... :	Bachelorarbeit Informatik Bachelorarbeit Softwaretechnik Bachelorarbeit Medieninformatik
19. Medienform:	
20. Angeboten von:	Soziokognitive Systeme

Modul: 81100 Bachelorarbeit Informatik

2. Modulkürzel:	050525002	5. Moduldauer:	Einsemestrig
3. Leistungspunkte:	12 LP	6. Turnus:	Wintersemester/ Sommersemester
4. SWS:	8	7. Sprache:	Deutsch/Englisch

8. Modulverantwortlicher:	Univ.-Prof. Dr. Stefan Funke
9. Dozenten:	Dozenten der Informatik
10. Zuordnung zum Curriculum in diesem Studiengang:	
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 Betreuer / Prüfer festgelegt
14. Literatur:	wird von Betreuer / Prüfer bekanntgegeben
15. Lehrveranstaltungen und -formen:	
16. Abschätzung Arbeitsaufwand:	Gesamtstunden: 360 h
17. Prüfungsnummer/n und -name:	81101 Bachelorarbeit Informatik (PL), Sonstige, Gewichtung: 1 Schriftliche Ausarbeitung zum vergebenen Thema sowie ein Vortrag über den Inhalt der bachelorarbeit
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Algorithmik