



Universität Stuttgart

Modulhandbuch
Studiengang Bachelor of Science Softwaretechnik
Prüfungsordnung: 2009

Universität Stuttgart
Keplerstr. 7
70174 Stuttgart

Inhaltsverzeichnis

Präambel	4
100 Basismodule	5
12060 Datenstrukturen und Algorithmen	6
17210 Einführung in die Softwaretechnik	8
10190 Mathematik für Informatiker und Softwaretechniker	10
10260 Programmierkurs	12
10280 Programmierung und Software-Entwicklung	14
16520 Software-Qualität	16
10940 Theoretische Grundlagen der Informatik	17
200 Kernmodule	19
11890 Algorithmen und Berechenbarkeit	20
14360 Einführung in die Technische Informatik	21
14370 Fachstudie Softwaretechnik	23
14390 Programmentwicklung	24
14480 Sichere und zuverlässige Softwaresysteme	25
16500 Software Engineering	26
16510 Software-Praktikum	27
210 Kernmodul Studienprojekt	28
16780 Studienprojekt-Th	29
300 Ergänzungsmodule	30
31400 English for Software Engineering	31
320 Katalog ISG	32
10140 Advanced Processor Architecture	33
10030 Architektur von Anwendungssystemen	35
10080 Datenbanken und Informationssysteme	37
10090 Grundlagen der Eingebetteten Systeme	39
10100 Grundlagen der Graphischen Ingenieursysteme	40
10110 Grundlagen der Künstlichen Intelligenz	42
10120 Grundlagen der Modellbildung und Simulation	44
10150 Grundlagen des Compilerbaus und der Programmiersprachen	46
10210 Mensch-Computer-Interaktion	48
10220 Modellierung	50
16790 Rechnerorganisation 1	52
10330 Systemkonzepte und -programmierung	54
11110 Verteilte Systeme	56
330 Katalog ISW	58
10140 Advanced Processor Architecture	59
10040 Bildsynthese	61
10050 Bildverstehen	63
10060 Computergraphik	65
10090 Grundlagen der Eingebetteten Systeme	67
10100 Grundlagen der Graphischen Ingenieursysteme	68
10110 Grundlagen der Künstlichen Intelligenz	70
10120 Grundlagen der Modellbildung und Simulation	72
10150 Grundlagen des Compilerbaus und der Programmiersprachen	74
14380 Hardware Verification and Quality Assessment	76
10170 Imaging Science	78

10180 Information Retrieval und Text Mining	80
10210 Mensch-Computer-Interaktion	81
10250 Parallele Systeme	83
10300 Rechnernetze	84
16790 Rechnerorganisation 1	86
10330 Systemkonzepte und -programmierung	88
11330 Visualisierung	90
310 Katalog SWT	92
10120 Grundlagen der Modellbildung und Simulation	93
10150 Grundlagen des Compilerbaus und der Programmiersprachen	95
10210 Mensch-Computer-Interaktion	97
16790 Rechnerorganisation 1	99
10330 Systemkonzepte und -programmierung	101
400 Schlüsselqualifikationen fachaffin	103
16610 Studienprojekt-Pr	104
700 Zusatzmodule	106
800 Schlüsselqualifikationen fachübergreifend gesamt	107
910 SQ Konto anerkannt	108

Präambel

nicht verfügbar

100 Basismodule

Zugeordnete Module:

- 12060 Datenstrukturen und Algorithmen
- 17210 Einführung in die Softwaretechnik
- 10190 Mathematik für Informatiker und Softwaretechniker
- 10260 Programmierkurs
- 10280 Programmierung und Software-Entwicklung
- 16520 Software-Qualität
- 10940 Theoretische Grundlagen der Informatik

Modul: 12060 Datenstrukturen und Algorithmen

2. Modulkürzel:	051510005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Erhard Plödereder		
9. Dozenten:	Stefan Funke		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 2. Semester → Basismodule		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051520005 Programmierung und Software-Entwicklung 		
12. Lernziele:	<p>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.</p> <p>Konkret:</p> <ul style="list-style-type: none"> • Kenntnis der Eigenschaften elementarer und häufig benötigter Algorithmen • Verständnis für die Auswirkungen theoretischer und tatsächlicher Komplexität • Erweiterung der Kompetenz im Entwurf und Verstehen von Algorithmen und der zugehörigen Datenstrukturen • Erste Begegnung mit nebenläufigen Algorithmen; sowohl „originär“ parallel, als auch parallelisierte Versionen bereits vorgestellter sequentieller Algorithmen 		
13. Inhalt:	<ul style="list-style-type: none"> • Vorgehensweise bei der Entwicklung und Implementierung von Algorithmen • Komplexität und Effizienz von Algorithmen, O-Notation • Wahl der Datenstrukturen; Listen, Bäume, Graphen; deren Definitionen, deren Datenstrukturen • diverse interne und externe Such- und Sortierverfahren (z.B. Linear-, Binär-, Interpolationssuche, AVL-, B-Bäume, internes und externes Hashing, mehrere langsame Sortierungen, Heap-, Quick-, Bucket-, Mergesort) • diverse Graphenalgorithmen (DFS, BFS, Besuchssequenzen, topol. Traversierung, Zusammenhangskomponenten, minimale Spannbäume, Dijkstra-, Floyd- kürzeste Wege) • Algorithmen auf Mengen und Relationen (transitive Hüllen, Warshall) • Korrektheitsbegriff und -formalismen; Spezifikation und Implementierung • Einige parallele und parallelisierte Algorithmen • einfache Elemente paralleler Programmierung, soweit für obiges notwendig 		
14. Literatur:	<ul style="list-style-type: none"> • Appelrath H.J., Ludewig. J., Skriptum Informatik, 1999 • Sedgewick, R., Algorithms in C, 1998 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 120601 Vorlesung Datenstrukturen und Algorithmen • 120602 Übung Datenstrukturen und Algorithmen 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	63 Stunden	

Nachbearbeitungszeit: 207 Stunden

17. Prüfungsnummer/n und -name: 12061 Datenstrukturen und Algorithmen (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein. Die genauen Details der Übungsleistungen und Ihrer Anrechnung werden zu Beginn der Lehrveranstaltung bekannt gegeben.

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 2. Semester
 - Basismodule
- BA (Komb) Informatik, 2. Semester
 - Module im Nebenfach
- B.Sc. Mathematik, 2. Semester
 - Nebenfach
 - Nebenfach Informatik
- B.Sc. Technische Kybernetik, 2. Semester
 - Ergänzungsmodule
 - Höhere Informatik
- B.Sc. Wirtschaftsinformatik, 2. Semester
 - Informatik (B 1)
- B.Sc. Mechatronik, 2. Semester
 - Kernmodule
- B.Sc. Maschinelle Sprachverarbeitung, 2. Semester
 - Basismodule
- B.Sc. Technikpädagogik, 2. Semester
 - Hauptfach Informatik
 - Basismodule Informatik
- B.Sc. Technikpädagogik, 2. Semester
 - Wahlpflichtfach
 - Informatik
- M.Sc. Technikpädagogik, 2. Semester
 - Wahlpflichtfach B
 - Affines Wahlpflichtfach Informatik
 - Affines Wahlpflichtfach Informatik Basismodule
- Allgemein Lehramt (GymPO I) ab PO 2010
 - Pflichtmodule

Modul: 17210 Einführung in die Softwaretechnik

2. Modulkürzel:	051520015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Jochen Ludewig		
9. Dozenten:	Jochen Ludewig		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 2. Semester → Basismodule		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen sowie entsprechende Programmiererfahrung		
12. Lernziele:	EST ist, wie der Name sagt, die allgemeine Einführung 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.		
13. Inhalt:	Die Vorlesung behandelt technische und andere Aspekte der Softwarebearbeitung, wie sie in der Praxis stattfindet. Die einzelnen Themen sind: <ul style="list-style-type: none"> • Abgrenzung und Motivation des Software Engineerings • Vorgehensmodelle; Software-Management; Software-Prüfung und Qualitätssicherung • Methoden, Sprachen und Werkzeuge für die einzelnen Phasen: Spezifikation, Grobentwurf, Feinentwurf, Codierung, Test		
14. Literatur:	Ludewig, Lichter: Software Engineering. dpunkt-Verlag, Heidelberg. 2. Aufl. 2010		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 172101 Vorlesung Einführung in die Softwaretechnik • 172102 Übung Einführung in die Softwaretechnik 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	17211 Einführung in die Softwaretechnik (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Vorleistung: Schein; keine Hilfsmittel zugelassen.		
18. Grundlage für ... :	<ul style="list-style-type: none"> • 16500 Software Engineering • 16510 Software-Praktikum 		
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:	B.Sc. Mechatronik, 5. Semester → Kernmodule B.Sc. Technikpädagogik, 4. Semester		

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

Modul: 10190 Mathematik für Informatiker und Softwaretechniker

2. Modulkürzel:	080300100	5. Moduldauer:	2 Semester
3. Leistungspunkte:	18.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	12.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Wolfgang Rump		
9. Dozenten:	Wolfgang Rump		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 1. Semester → Basismodule		
11. Voraussetzungen:	Keine, die Teilnahme an einem Mathematik Vorkurs wird empfohlen.		
12. Lernziele:	Die Studierenden haben die mathematischen Grundlagen für die Studiengänge Informatik bzw. Softwaretechnik erarbeitet und den selbständigen und kreativen Umgang mit den mathematischen Stoffgebieten gelernt.		
13. Inhalt:	<p>1. Semester:</p> <ul style="list-style-type: none"> • 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) <p>2. Semester:</p> <ul style="list-style-type: none"> • Differential- und Integralrechnung (Funktionen einer und mehrerer Variablen, Ableitungen, Taylorentwicklungen, Extremwerte, Integration, Anwendungen) • Gewöhnliche Differentialgleichungen (elementar lösbare Differentialgleichungen, Existenz und Eindeutigkeit von Lösungen) 		
14. Literatur:	<ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none"> • 101901 Vorlesung Mathematik 1 für Informatik und Softwaretechnik • 101902 Übung Mathematik 1 für Informatik und Softwaretechnik • 101903 Vorlesung Mathematik 2 für Informatik und Softwaretechnik • 101904 Übung Mathematik 2 für Informatik und Softwaretechnik 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 126 Stunden Nachbearbeitungszeit: 414 Stunden		
17. Prüfungsnummer/n und -name:	10191 Mathematik für Informatiker und Softwaretechniker (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Ein Übungsschein aus den beiden Veranstaltungen, jeweils im 1. oder 2. Fachsemester zu erwerben		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 1. Semester
 - Basismodule
- B.Sc. Technikpädagogik, 1. Semester
 - Hauptfach Informatik
 - Basismodule Informatik
- Allgemein Lehramt (GymPO I) ab PO 2010
 - Pflichtmodule

Modul: 10260 Programmierkurs

2. Modulkürzel:	051520010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Jochen Ludewig		
9. Dozenten:	<ul style="list-style-type: none"> • Sascha Riexinger • Holger Röder 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 1. Semester → Basismodule		
11. Voraussetzungen:	Keine		
12. Lernziele:	Selbstständiges Erstellen von Programmen und Lösung von Programmieraufgaben in einer vorgegebenen Programmiersprache wie Java.		
13. Inhalt:	<p>Der Programmierkurs ergänzt die Vorlesung Programmierung und Software-Entwicklung (PSE). Die Teilnehmer erlernen eine weitere Programmiersprache (Java). Ihre Merkmale, Syntax und Semantik, werden denen der in PSE gelehrt Sprache gegenübergestellt. Praktische Übungen bereiten die Teilnehmer auf die Bearbeitung der Schein-Aufgabe vor.</p> <p>Die Lehrveranstaltung findet in zwei Varianten statt. Die Teilnahme richtet sich nach dem Studiengang:</p> <p>S. Riexinger:</p> <ul style="list-style-type: none"> • BSc. Informatik • BA (Komb) Informatik • BSc. Maschinelle Sprachverarbeitung <p>H. Röder:</p> <ul style="list-style-type: none"> • BSc. Softwaretechnik • BSc. Wirtschaftsinformatik • BSc. Technikpädagogik • MSc. Technikpädagogik 		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	102601 Übung Programmierkurs		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 21 Stunden Nachbearbeitungszeit: 69 Stunden		
17. Prüfungsnummer/n und -name:	10261 Programmierkurs (USL), Sonstiges, Gewichtung: 1.0, Übungsschein - Scheinkriterien werden zu Beginn der Veranstaltung angekündigt.		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

21. Zuordnung zu weiteren Curricula:
- B.Sc. Informatik, 1. Semester
 - Basismodule
 - BA (Komb) Informatik, 1. Semester
 - Module im Nebenfach
 - B.Sc. Wirtschaftsinformatik, 1. Semester
 - Informatik (B 1)
 - B.Sc. Maschinelle Sprachverarbeitung, 1. Semester
 - Basismodule
 - B.Sc. Technikpädagogik, 1. Semester
 - Hauptfach Informatik
 - Basismodule Informatik
 - B.Sc. Technikpädagogik, 1. Semester
 - Wahlpflichtfach
 - Informatik
 - M.Sc. Technikpädagogik, 1. Semester
 - Wahlpflichtfach B
 - Affines Wahlpflichtfach Informatik
 - Affines Wahlpflichtfach Informatik Basismodule
-

Modul: 10280 Programmierung und Software-Entwicklung

2. Modulkürzel:	051520005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Bernhard Mitschang		
9. Dozenten:	Bernhard Mitschang		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 1. Semester → Basismodule		
11. Voraussetzungen:	Keine. Teilnahme an einem Mathematik Vorkurs wird empfohlen.		
12. Lernziele:	<p>Die Teilnehmer haben die wichtigsten Konzepte einer höheren Programmiersprache und ihrer Verwendung verstanden und sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen) zu analysieren und selbst zu konzipieren und zu implementieren. Sie kennen die Möglichkeiten, Daten- und Ablaufstrukturen zu entwerfen, zu beschreiben und zu codieren. Sie haben die Abstraktionskonzepte moderner Programmiersprachen verstanden. Sie kennen die Techniken und Notationen zur Definition kontextfreier Programmiersprachen und können damit arbeiten.</p>		
13. Inhalt:	<ul style="list-style-type: none"> • Algorithmen und funktionale Programmierung Kap. 1 verwendet nur die funktionale Teilmenge der Programmiersprache Ada, keine Variablen, keine Prozeduren. Grammatik, Formale Sprachen und BNF werden eingeführt. • Imperative Programmierung Kap. 2 erweitert die verwendete Sprache durch die prozeduralen Konzepte, also Variablen und Prozeduren. Zu den Sprachkonstrukten werden Vor- und Nachbedingungen, mit den Schleifen die Invarianten eingeführt. Datentypen werden schrittweise ausgebaut. In Zusammenhang mit den Zeigern werden die Konzepte für Keller und Halde vermittelt. Die Entwicklung einfacher Programme wird gezeigt und geübt. • Aufbau und Organisation komplexer Programme. Die Modularisierung, die bei größeren Programmen notwendig ist, führt zur Kapselung und zu den abstrakten Datentypen. Damit entsteht die Möglichkeit, neue Datenstrukturen und Datentypen sicher zu definieren. Die Konzepte der Kompilation und der Interpretation werden erläutert. Wichtige Beispiele komplexer Datentypen werden entwickelt. Die Konzepte der Generalisierung (generische Einheiten) werden vermittelt. • Ausnahmebehandlung Möglichkeiten und Probleme der Ausnahmebehandlung sind Gegenstände dieses kurzen Kapitels. • Objektorientierte Programmierung Am Ende des Semesters steht ein Ausblick in die objektorientierte Programmierung, d.h. die Umsetzung der bereits bekannten Konzepte (ADTs) in die objektorientierte Sichtweise und die Vererbung. Dieser Teil bereitet die Programmierung in einer objektorientierten Sprache (3. Semester) vor. 		
14. Literatur:	<ul style="list-style-type: none"> • Manuskripte: V.Claus (WS 08/09 bis SS 2009) • Appelrath, Hans-Jürgen und Ludewig, Jochen, "Skriptum Informatik - eine konventionelle Einführung", Verlag der Fachvereine Zürich und B.G. Teubner Stuttgart, 4. Auflage 1999 		

	<ul style="list-style-type: none"> • Nagl., M., "Softwaretechnik mit Ada 95. Entwicklung großer Systeme.", Vieweg-Verlag, Wiesbaden 1999 • Barnes, J.G.P., "Programming in Ada 95", 2. Auflage, Addison-Wesley 1998
--	---

15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 102801 Vorlesung Programmierung und Softwareentwicklung • 102802 Übung Programmierung und Softwareentwicklung
--------------------------------------	--

16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 63 Stunden Vor-/Nachbearbeitungszeit: 187 Stunden Prüfungsvorbereitung: 20 Stunden
---------------------------------	---

17. Prüfungsnummer/n und -name:	10281 Programmierung und Software-Entwicklung (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Studienleistung: Übungsschein. Voraussetzungen werden zu Beginn vom Dozenten festgesetzt. Dazu gehören eine bestimmte Anzahl von Vorträgen in den Übungen und ein bestimmter Teil der Übungspunkte. Modulprüfung: schriftlich, 120 Minuten, keine Hilfsmittel
---------------------------------	--

18. Grundlage für ... :	
-------------------------	--

19. Medienform:	
-----------------	--

20. Angeboten von:	
--------------------	--

21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 1. Semester → Basismodule BA (Komb) Informatik, 1. Semester → Module im Nebenfach B.Sc. Wirtschaftsinformatik, 1. Semester → Informatik (B 1) B.Sc. Mechatronik, 1. Semester → Kernmodule B.Sc. Maschinelle Sprachverarbeitung, 1. Semester → Basismodule B.Sc. Technikpädagogik, 1. Semester → Hauptfach Informatik → Basismodule Informatik B.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach → Informatik M.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Basismodule B.Sc. Simulation Technology, 1. Semester → Grundstudium Allgemein Lehramt (GymPO I) ab PO 2010, 1. Semester → Pflichtmodule
--------------------------------------	---

Modul: 16520 Software-Qualität

2. Modulkürzel:	051520105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Jochen Ludewig		
9. Dozenten:	<ul style="list-style-type: none"> • Jochen Ludewig • Ivan Bogicevic 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 1. Semester → Basismodule		
11. Voraussetzungen:	<ul style="list-style-type: none"> • gleichzeitiger Besuch der Einführungsvorlesung und des Programmierkurses 		
12. Lernziele:	Die Teilnehmer kennen und verstehen den Begriff der Software-Qualität. Sie kennen Techniken, deren Anwendung zu einer guten Software-Qualität beiträgt, und können sie anwenden.		
13. Inhalt:	In der Vorlesung wird der Begriff der Software-Qualität vermittelt und am Beispiel anschaulich gemacht. In der Übung wird ein großes Softwaresystem bearbeitet.		
14. Literatur:	<ul style="list-style-type: none"> • Ludewig J., Lichter, H., Software Engineering - Grundlagen, Menschen, Prozesse, Techniken, 2007 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 165201 Vorlesung Software-Qualität • 165202 Übung Software-Qualität 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 21 Stunden Nachbearbeitungszeit: 69 Stunden		
17. Prüfungsnummer/n und -name:	16521 Software-Qualität (USL), Studienbegleitend, Gewichtung: 1.0, Übungsschein; Scheinkriterien werden zu Beginn der Veranstaltung angekündigt.		
18. Grundlage für ... :	17210 Einführung in die Softwaretechnik		
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:			

Modul: 10940 Theoretische Grundlagen der Informatik

2. Modulkürzel:	050420005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Volker Diekert		
9. Dozenten:	<ul style="list-style-type: none"> • Ulrich Hertrampf • Volker Diekert 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 1. Semester → Basismodule		
11. Voraussetzungen:	Keine		
12. Lernziele:	<ul style="list-style-type: none"> • Logik und Diskrete Strukturen: Die Studierenden haben die grundsätzlichen Kenntnisse in Logik und Diskreter Mathematik erworben, wie sie in den weiteren Grundvorlesungen der Informatik in verschiedenen Bereichen benötigt werden. • Automaten und Formale Sprachen: Die Studierenden beherrschen wichtige theoretische Grundlagen der Informatik, insbesondere die Theorie und Algorithmik endlicher Automaten. Hierzu gehört das Kennenlernen, Einordnung und Trennung der Chomskyschen Sprachklassen. 		
13. Inhalt:	<ul style="list-style-type: none"> • Logik und Diskrete Strukturen: Einführung in die Aussagenlogik; formale Sprache; Semantik (Wahrheitswerte); Syntax (Axiome und Schlussregeln); Normalformen; Hornformeln; aussagenlogische Resolution; Korrektheit und Vollständigkeit für die Aussagenlogik; Einführung in die Prädikatenlogik 1. Stufe; formale Sprache; Semantik und Syntax; Normalformen; Herbrand-Theorie; prädikatenlogische Resolution; Kombinatorik, Graphen, elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, RSA-Verfahren. • Automaten und Formale Sprachen: Deterministische- bzw. nichtdeterministische endliche Automaten, reguläre Ausdrücke, Minimierung endlicher Automaten, Iterationslemmata für reguläre und kontextfreie Sprachen, Normalformen, Kellerautomaten, Lösen des Wortproblems kontextfreier Sprachen mit dem CYK-Algorithmus, linear beschränkte Automaten, kontextsensitive Grammatiken, Typ 0-Grammatiken und Turingmaschinen. 		
14. Literatur:	<ul style="list-style-type: none"> • John Hopcroft, Jeffrey Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988 • Uwe Schöning, Theoretische Informatik - kurzgefasst, 1999 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 109401 Vorlesung Logik und Diskrete Strukturen • 109402 Übung Logik und Diskrete Strukturen • 109403 Vorlesung Automaten und Formale Sprachen • 109404 Übung Automaten und Formale Sprachen 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	84 Stunden	

Nachbearbeitungszeit: 276 Stunden

17. Prüfungsnummer/n und -name: 10941 Theoretische Grundlagen der Informatik (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 1. Semester
 - Basismodule
- B.Sc. Maschinelle Sprachverarbeitung, 1. Semester
 - Basismodule
- B.Sc. Technikpädagogik, 1. Semester
 - Hauptfach Informatik
 - Kernmodule Informatik
- B.Sc. Technikpädagogik, 1. Semester
 - Wahlpflichtfach
 - Informatik
- M.Sc. Technikpädagogik, 1. Semester
 - Wahlpflichtfach B
 - Affines Wahlpflichtfach Informatik
 - Affines Wahlpflichtfach Informatik Kernmodule
- Allgemein Lehramt (GymPO I) ab PO 2010, 1. Semester
 - Pflichtmodule

200 Kernmodule

Zugeordnete Module:

- 11890 Algorithmen und Berechenbarkeit
- 14360 Einführung in die Technische Informatik
- 14370 Fachstudie Softwaretechnik
- 14390 Programmentwicklung
- 14480 Sichere und zuverlässige Softwaresysteme
- 16500 Software Engineering
- 16510 Software-Praktikum

Modul: 11890 Algorithmen und Berechenbarkeit

2. Modulkürzel:	050420020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Stefan Funke		
9. Dozenten:	<ul style="list-style-type: none"> • Stefan Funke • Volker Diekert • Ulrich Hertrampf 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 3. Semester → Kernmodule		
11. Voraussetzungen:	Vorlesungen aus dem 1. und 2. Semester		
12. Lernziele:	Die Studierenden kennen die Klassifizierung von Algorithmen in effizient berechenbar, NP-vollständig, PSPACE-Algorithmen und prinzipielle Unberechenbarkeit. Sie haben wichtige Entwurfstrategien und Analysemethoden kennengelernt.		
13. Inhalt:	Berechenbarkeit vs. Unberechenbarkeit, Church These, NP-Vollständigkeit, PSPACE-Algorithmen (QBF). Entwurfstrategien: Teile und Beherrsche, gierig (greedy), Dynamisches Programmieren, Randomisierte Algorithmen		
14. Literatur:	<ul style="list-style-type: none"> • John Hopcroft, Jeffrey Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988 • Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms (Second Edition), 2001 • Volker Diekert, Entwurf und Analyse effizienter Algorithmen (Vorlesungsskript), 2006 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 118901 Vorlesung Algorithmen und Berechenbarkeit • 118902 Übung Algorithmen und Berechenbarkeit 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiumszeit /	138 h	
	Nacharbeitszeit:		
	Gesamt:	180 h	
17. Prüfungsnummer/n und -name:	11891 Algorithmen und Berechenbarkeit (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:	B.Sc. Technikpädagogik, 5. Semester → Hauptfach Informatik → Pflichtmodule Informatik M.Sc. Technikpädagogik, 5. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Pflichtmodule		

Modul: 14360 Einführung in die Technische Informatik

2. Modulkürzel:	051400105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Otto Eggenberger		
9. Dozenten:	Otto Eggenberger		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 3. Semester → Kernmodule		
11. Voraussetzungen:	Keine		
12. Lernziele:	Der/die Studierende kennt die grundlegende Funktionsweise eines Computers, versteht die elektrotechnischen Grundlagen und Technologien und kann einfache digitale Schaltungen analysieren, entwerfen und optimieren.		
13. Inhalt:	<p>Grundlegende Funktionsweise eines Computers</p> <ul style="list-style-type: none"> • Informationsdarstellung • Zahlendarstellung und Codes • Digitale Grundbausteine • Logische Funktionen, Speicherelemente • Befehlsausführung, Programmablauf <p>Elektrotechnische Grundlagen</p> <ul style="list-style-type: none"> • Physikalische Grundbegriffe • Elektrische Spannung, elektrischer Strom • Elektrische Netzwerke • Halbleiterbauelemente • Digitale Grundschaltungen <p>Digitale Schaltungen</p> <ul style="list-style-type: none"> • Schaltnetzwerke • Boolesche Algebra und Schaltalgebra • Darstellung und Minimierung von Schaltfunktionen • Rückkopplung, Zustandsbegriff • Automaten und sequentielle Netzwerke • Digitale Standardschaltungen • Entwurfsmethodik 		
14. Literatur:	<p>Dirk W. Hoffman: Grundlagen der technischen Informatik, Hanser, 2007 Bernd Becker, Rolf Drechsler, Paul Molitor: Technische Informatik, Pearson Studium, 2005 Jörg Keller, Wolfgang J. Paul: Hardware Design, Teubner, 3. Aufl. 2005</p>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 143601 Vorlesung Einführung in die Technische Informatik • 143602 Gruppenübungen Einführung in die Technische Informatik 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	
	Nachbearbeitungszeit:	126 Stunden	
17. Prüfungsnummer/n und -name:	14361 Einführung in die Technische Informatik (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von:

21. Zuordnung zu weiteren Curricula: BA (Komb) Informatik, 3. Semester
 → Module im Nebenfach
 Allgemein Lehramt (GymPO I) ab PO 2010
 → Pflichtmodule

Modul: 14370 Fachstudie Softwaretechnik

2. Modulkürzel:	051520185	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Jochen Ludewig		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 6. Semester → Kernmodule		
11. Voraussetzungen:	<ul style="list-style-type: none"> Bestandene Prüfung "Software Engineering" 		
12. Lernziele:	Die Teilnehmer sind in der Lage, eine konkrete praktische Frage der Softwaretechnik, beispielsweise über die anzuwendende Methode oder das geeignete Werkzeug, zu analysieren und zu entscheiden und ihre Entscheidung angemessen zu präsentieren. Die Arbeit erfolgt in Dreiergruppen.		
13. Inhalt:	Eine Gruppe analysiert eine (im Allgemeinen aus der Praxis kommende) Frage auf der Basis der Literatur und eigener Untersuchungen, auch Befragungen, und präsentiert ihre Empfehlung mündlich und in Form eines Berichts.		
14. Literatur:	-		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> 143701 Praktikum Fachstudie Softwaretechnik 143702 Teamarbeit an den beteiligten Instituten mit örtlicher fachlicher Betreuung 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 126 Stunden		
17. Prüfungsnummer/n und -name:	14371 Fachstudie Softwaretechnik (USL), Studienbegleitend, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:			

Modul: 14390 Programmentwicklung

2. Modulkürzel:	051520120	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Jochen Ludewig		
9. Dozenten:	<ul style="list-style-type: none"> • Jochen Ludewig • Holger Röder 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 3. Semester → Kernmodule		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Programmierkurs • 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:	<ul style="list-style-type: none"> • Grundlagen der objektorientierten Programmierung • Spezifikation und Entwurf objektorientierter Programme mit UML • Programmierung in Java 		
14. Literatur:	<ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none"> • 143901 Vorlesung Programmentwicklung • 143902 Übung Programmentwicklung 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 126 Stunden		
17. Prüfungsnummer/n und -name:	14391 Programmentwicklung (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Klausur 60 min, keine Vorleistungen.		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:			

Modul: 14480 Sichere und zuverlässige Softwaresysteme

2. Modulkürzel:	051520115	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:			
9. Dozenten:			
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 5. Semester → Kernmodule		
11. Voraussetzungen:	<ul style="list-style-type: none"> Teilnahme an der Lehrveranstaltung "Einführung in die Softwaretechnik" 		
12. Lernziele:	Die Teilnehmer beherrschen Techniken, deren Anwendung die Zuverlässigkeit der Software verbessert oder garantiert. Dabei stehen formale Ansätze im Vordergrund.		
13. Inhalt:	<ul style="list-style-type: none"> Notationen und Verfahren zur formalen Beschreibung und Prüfung der Software 		
14. Literatur:	-		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> 144801 Vorlesung Sichere und zuverlässige Softwaresysteme 144802 Übung Sichere und zuverlässige Softwaresysteme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	14481 Sichere und zuverlässige Softwaresysteme (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:			

Modul: 16500 Software Engineering

2. Modulkürzel:	051520110	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	-
8. Modulverantwortlicher:	Jochen Ludewig		
9. Dozenten:	Jochen Ludewig		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Kernmodule		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Einführung in die Softwaretechnik • Programmentwicklung 		
12. Lernziele:	Die Teilnehmer haben tiefe und umfassende Kenntnisse auf dem Gebiet des Softwareprojekt-Managements und in den Techniken der Software-Bearbeitung.		
13. Inhalt:	<p>Ergänzend zur "Einführung in die Softwaretechnik" und daran anknüpfend behandelt diese Lehrveranstaltung folgende Themen:</p> <ul style="list-style-type: none"> • Geschichte des Software Engineerings • Organisationsaspekte der Software-Bearbeitung • Software-Prozesse, Prozess-Bewertung und -Verbesserung • Software-Wartung • Weitere ausgewählte Kapitel des Software Engineerings 		
14. Literatur:	<ul style="list-style-type: none"> • Ludewig J., Lichter, H., Software Engineering - Grundlagen, Menschen, Prozesse, Techniken, 2. Aufl. 2007 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 165001 Vorlesung Software Engineering • 165002 Übung Software Engineering 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	16501 Software Engineering (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Keine Hilfsmittel zugelassen.		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Softwaretechnologie		
21. Zuordnung zu weiteren Curricula:			

Modul: 16510 Software-Praktikum

2. Modulkürzel:	051520180	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	-
8. Modulverantwortlicher:	Jochen Ludewig		
9. Dozenten:	Ivan Bogicevic		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 3. Semester → Kernmodule		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Einführung in die Softwaretechnik • gleichzeitiger Besuch der Programmentwicklung 		
12. Lernziele:	Die Teilnehmer können eine Software-Entwicklung von der Spezifikation bis zur Auslieferung durchführen.		
13. Inhalt:	Die Teilnehmer bearbeiten in Dreiergruppen eine zentral gestellte Aufgabe. Sie erheben dazu die notwendigen Informationen, erstellen die notwendigen Dokumente und implementieren ein Programm, das die Aufgabe löst.		
14. Literatur:	<ul style="list-style-type: none"> • Ludewig J., Lichter, H., Software Engineering - Grundlagen, Menschen, Prozesse, Techniken. dpunkt-Verlag, Heidelberg, 2. Aufl. 2010 		
15. Lehrveranstaltungen und -formen:	165101 Praktikum Software-Praktikum		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 126 Stunden		
17. Prüfungsnummer/n und -name:	16511 Software-Praktikum (USL), Studienbegleitend, Gewichtung: 1.0		
18. Grundlage für ... :	<ul style="list-style-type: none"> • 16610 Studienprojekt-Pr • 16780 Studienprojekt-Th 		
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:			

210 Kernmodul Studienprojekt

Zugeordnete Module: 16780 Studienprojekt-Th

Modul: 16780 Studienprojekt-Th

2. Modulkürzel:	051520192	5. Moduldauer:	2 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	unregelmäßig
4. SWS:	6.0	7. Sprache:	-
8. Modulverantwortlicher:	Jochen Ludewig		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Kernmodul Studienprojekt		
11. Voraussetzungen:	Die Module Einführung in die Softwaretechnik, Programmentwicklung und Software-Praktikum müssen vor Beginn des Studienprojekts absolviert sein. Das Studienprojekt-Th bildet mit dem Studienprojekt-Pr eine Einheit; beide können nur zusammen begonnen werden. Die Vorleistungen (Scheine) aus dem Studienprojekt-Pr sind für die abschließende Prüfung des Studienprojekt-Th Voraussetzung.		
12. Lernziele:	Vorlesung und Seminar dienen dazu, theoretische Grundlagen zum Studienprojekt-Pr zu vermitteln und die Arbeit im Projekt zu reflektieren.		
13. Inhalt:	Die konkreten Themen und Inhalte hängen vom jeweiligen Projekt ab.		
14. Literatur:	<ul style="list-style-type: none"> • Deininger, Lichter, Ludewig, Schneider, Studien-Arbeiten, 5. Aufl. 2005 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 167801 Vorlesung Studienprojekt-Th • 167802 Seminar Studienprojekt-Th 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 63 Stunden Nachbearbeitungszeit: 189 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 16781 Studienprojekt-Th - Prüfung (PL), schriftlich oder mündlich, Gewichtung: 2.0 • 16782 Studienprojekt-Th - Schein (LBP), schriftlich oder mündlich, 25 Min., Gewichtung: 1.0 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:			

300 Ergänzungsmodule

Zugeordnete Module: 31400 English for Software Engineering
 320 Katalog ISG
 330 Katalog ISW
 310 Katalog SWT

Modul: 31400 English for Software Engineering

2. Modulkürzel:	051900012	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:			
9. Dozenten:			
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule		
11. Voraussetzungen:			
12. Lernziele:			
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:			
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	31401 English for Software Engineering (USL), schriftlich, eventuell mündlich, 30 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:			

320 Katalog ISG

Zugeordnete Module:	10140	Advanced Processor Architecture
	10030	Architektur von Anwendungssystemen
	10080	Datenbanken und Informationssysteme
	10090	Grundlagen der Eingebetteten Systeme
	10100	Grundlagen der Graphischen Ingenieursysteme
	10110	Grundlagen der Künstlichen Intelligenz
	10120	Grundlagen der Modellbildung und Simulation
	10150	Grundlagen des Compilerbaus und der Programmiersprachen
	10210	Mensch-Computer-Interaktion
	10220	Modellierung
	16790	Rechnerorganisation 1
	10330	Systemkonzepte und -programmierung
	11110	Verteilte Systeme

Modul: 10140 Advanced Processor Architecture

2. Modulkürzel:	051700010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> • Hans-Joachim Wunderlich • Stefan Holst 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051700005 Rechnerorganisation 		
12. Lernziele:	Good understanding of the basic concepts used in modern CPUs and computing systems. Awareness of the challenges in modern processor design and the reasoning behind current and future design trends.		
13. Inhalt:	Classic topics in computer architecture as hardware/software interface are discussed as well as more advanced topics which include: <ul style="list-style-type: none"> • Technology basics: Design patterns, fabrication, yield, test and reliability, cost and quality, scaling. • Performance: Frequency and instructions per clock cycle, performance analysis and optimization. • Power dissipation: Analysis and optimization of power and performance, power and scaling. • Computer arithmetic: Efficient hardware for basic arithmetic, implementation of exponential, logarithm and trigonometric functions, floating point arithmetic and standards, arithmetic pipelines and filter, real-world floating point implementations like the Cell SPE or SPARC. • Instruction parallelism: Super scalar computing, static and dynamic scheduling, out-of-order execution, VLIW-processors, multithreading • Parallel architectures: Shared memory and message passing, multi-core processors, multi-core systems on a chip and emerging many-core technologies found in current graphic accelerators • Memory hierarchy: Memory technology and cache design. • Fault tolerance for single processors and multi processor systems 		
14. Literatur:	<ul style="list-style-type: none"> • I. Koren, Computer Arithmetic Algorithms, 2001 • J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 2006 • S. Iman, M. Pedram, Logic Synthesis for Low Power VLSI Designs, 1997 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101401 Vorlesung Grundlagen der Rechnerarchitektur 		

• 101402 Übung Grundlagen der Rechnerarchitektur

16. Abschätzung Arbeitsaufwand: Präsenzzeit: 42 Stunden
Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name: 10141 Advanced Processor Architecture (PL), schriftliche Prüfung,
90 Min., Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula: B.Sc. Informatik, 3. Semester
→ Ergänzungsmodule
→ Katalog ISG 1-3
BA (Komb) Informatik, 5. Semester
→ Module im Nebenfach
→ Katalog ISG
Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
→ Ergänzendes Modul
Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
→ Wahlmodule

Modul: 10030 Architektur von Anwendungssystemen

2. Modulkürzel:	052010002	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISG		
11. Voraussetzungen:	Vorlesungen des Grundstudiums.		
12. Lernziele:	Am Ende des Moduls ist das Konzept einer Anwendungsarchitektur verstanden. Die Rolle von Middleware im Rahmen von Anwendungsarchitekturen ist klar. Grundsätzliche Strukturen und Muster in Anwendungsarchitekturen sind bekannt. Nicht-funktionale Eigenschaften von Anwendungssystemen und deren Bedeutung sind verstanden.		
13. Inhalt:	<ul style="list-style-type: none"> • "Anwendungsarchitektur": Terminologie, Rollen & Artefakte • Datenbanksysteme • Schichtenarchitekturen (Client-Server Cut, N-Tier) • API, RPC & Middleware • Transaktionen • Nachrichtenorientierung • Message-Oriented Middleware • TP Monitore (direct TP, queued TP) • QoS (high availability, scalability, security,...) • Komponenten & ihre Beschreibung (EJB & WSDL) • Application Server (JEE) • Architekturstile (POSA,...) • Lose Kopplung & SOA 		
14. Literatur:	<ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none"> • 100301 Vorlesung Grundlagen der Architektur von Anwendungssystemen 		

- 100302 Übung Grundlagen der Architektur von Anwendungssystemen

16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10031 Architektur von Anwendungssystemen (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISG

Modul: 10080 Datenbanken und Informationssysteme

2. Modulkürzel:	051200025	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Bernhard Mitschang		
9. Dozenten:	<ul style="list-style-type: none"> • Bernhard Mitschang • Holger Schwarz 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 052010001 Modellierung 		
12. Lernziele:	Die Studierenden haben das erforderliche Verständnis und die Kenntnisse für die Implementierung von Datenbanksystemen erworben.		
13. Inhalt:	<p>Diese Vorlesung ist als Einstiegsvorlesung für das Vertiefungsgebiet Datenbanksysteme konzipiert. Es wird dabei versucht, aufbauend auf dem Inhalt der Vorlesung Modellierung, die wesentlichen Themen aus dem Bereich Datenbanksysteme zu vertiefen, insbesondere unter Berücksichtigung von Implementierungskonzepten und -techniken.</p> <p>Übersicht zur Stoffauswahl:</p> <ul style="list-style-type: none"> • Datenbanksystemarchitektur und Implementierungskonzepten • Implementierung relationaler Anfragesprachen (Anfrageverarbeitung und Anfrageoptimierung) • Transaktionsverarbeitung <p>Stoffauswahl, -umfang und Detaillierungsgrad werden aus der Sicht der Datenbanksystemimplementierung getroffen.</p>		
14. Literatur:	<ul style="list-style-type: none"> • A. Kemper, A. Eickler, Datenbanksysteme - Eine Einführung, 2004 • T. Härder, E. Rahm, Datenbanksysteme - Konzepte und Techniken der Implementierung, 2001 • H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems. The Complete Book, 2003 • R. Elmasri, S. Navathe, Fundamentals of Database Systems, 2003 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 100801 Vorlesung Datenbanken und Informationssysteme • 100802 Übung Datenbanken und Informationssysteme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10081 Datenbanken und Informationssysteme (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Modalitäten werden in der ersten Vorlesung angegeben		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 5. Semester → Ergänzungsmodule		

→ Katalog ISG 1-3

BA (Komb) Informatik, 5. Semester

→ Module im Nebenfach

→ Katalog ISG

B.Sc. Maschinelle Sprachverarbeitung, 5. Semester

→ Wahlbereich E/I

Allgemein Lehramt (GymPO I) ab PO 2010, 5. Semester

→ Ergänzendes Modul

Allgemein Lehramt (GymPO I) ab PO 2010, 5. Semester

→ Wahlmodule

Modul: 10090 Grundlagen der Eingebetteten Systeme

2. Modulkürzel:	051711010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051700005 Rechnerorganisation 		
12. Lernziele:	Methodische und werkzeunterstützte Konstruktion von Hardware und Software für auf spezifische Anwendungen hin optimierte eingebettete Systeme.		
13. Inhalt:	Modelle zur Systemspezifikation, Modellierungssprachen und ihre Simulation, System- und Architektursynthese, Allokation von Ressourcen und Bindung von Aufgaben/Operationen, Hardware-Software-Partitionierung, Verfahren zur Ablaufplanung für parallele Architekturen, Optimierungsverfahren, Anwendungsspezifische Prozessoren, On-Chip/Board-Verbindungsnetzwerke.		
14. Literatur:	<ul style="list-style-type: none"> • J. Teich, Digitale Hardware/Software-Systeme, 2. Auflage, 2007 • P. Marwedel, Embedded System Design, 2006 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 100901 Vorlesung Grundlagen der Eingebetteten Systeme • 100902 Übung Grundlagen der Eingebetteten Systeme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 10091 Grundlagen der Eingebetteten Systeme (PL), schriftlich, eventuell mündlich, Gewichtung: 75.0 • 10092 Grundlagen der Eingebetteten Systeme - Rechnerübungen (LBP), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 25.0 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG		

Modul: 10100 Grundlagen der Graphischen Ingenieursysteme

2. Modulkürzel:	051400005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dieter Roller		
9. Dozenten:	Dieter Roller		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	Grundstudium		
12. Lernziele:	<ul style="list-style-type: none"> • Kenntnis und Verständnis von Modellen bei der Produktentwicklung • Grundkenntnisse über die wichtigsten Modellarten, Algorithmen und Datenstrukturen und Techniken für den Datenaustausch 		
13. Inhalt:	Inhalte: <ul style="list-style-type: none"> • Anforderungen an CAD-Systeme • zweidimensionale Modelle • dreidimensionale Modelle • interaktive Modellerstellung • Einführung in die Modifikationstechnik u. parametrische Modellierung • Methoden zur Modellmodifikation • Grundlagen der parametrischen Modellierung • Ansätze und Verfahren zur parametrischen Variantenerzeugung • Ausgewählte Anwendungsbeispiele • Überblick über weitergehende Modellieransätze • Datenverwaltung in CAD 		
14. Literatur:	<ul style="list-style-type: none"> • D. Roller, CAD - Effiziente Anpassungs- und Variantenkonstruktion, Springer-Verlag • Literatur, siehe Webseite zur Veranstaltung 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101001 Vorlesung Grundlagen der Graphischen Ingenieursysteme • 101002 Übung Grundlagen der Graphischen Ingenieursysteme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	
	Nachbearbeitungszeit:	138 Stunden	
17. Prüfungsnummer/n und -name:	10101 Grundlagen der Graphischen Ingenieursysteme (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 4. Semester → Module im Nebenfach		

- Katalog ISG
 - Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Ergänzendes Modul
 - Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Wahlmodule
-

Modul: 10110 Grundlagen der Künstlichen Intelligenz

2. Modulkürzel:	051900205	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Gunther Heidemann		
9. Dozenten:	Gunther Heidemann		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 080300100 Mathematik für Informatiker und Softwaretechniker 		
12. Lernziele:	Der Student / die Studentin beherrscht die Grundlagen der Künstlichen Intelligenz, kann Probleme der KI selbständig einordnen und mit den erlernten Methoden und Algorithmen bearbeiten.		
13. Inhalt:	<ul style="list-style-type: none"> • Intelligenz • Agentenbegriff • Problemlösen durch Suchen, Suchverfahren • Probleme mit Rand- und Nebenbedingungen • Spiele • Aussagen- und Prädikatenlogik • Logikbasierte Agenten, Wissensrepräsentation • Inferenz • Planen • Unsicherheit, probabilistisches Schließen • Probabilistisches Schließen über die Zeit • Sprachverarbeitung • Entscheidungstheorie • Lernen 		
14. Literatur:	<ul style="list-style-type: none"> • G. F. Luger, Künstliche Intelligenz, 2001 • S. Russell, P. Norvig, Künstliche Intelligenz, 2004 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101101 Vorlesung Grundlagen der Künstlichen Intelligenz • 101102 Übung Grundlagen der Künstlichen Intelligenz 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	10111 Grundlagen der Künstlichen Intelligenz (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3		

- BA (Komb) Informatik, 5. Semester
 - Module im Nebenfach
 - Katalog ISG
 - B.Sc. Wirtschaftsinformatik, 5. Semester
 - Informatik (B 1)
 - B.Sc. Wirtschaftsinformatik, 5. Semester
 - Wahlpflichtbereich (Bereich C)
 - Wahlbereich II: Informatik Basis
 - B.Sc. Wirtschaftsinformatik, 5. Semester
 - Wahlpflichtbereich (Bereich C)
 - Wahlbereich III: Informatik Simulation
 - B.Sc. Maschinelle Sprachverarbeitung, 5. Semester
 - Wahlbereich E/I
-

Modul: 10120 Grundlagen der Modellbildung und Simulation

2. Modulkürzel:	051240010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Marc Alexander Schweitzer		
9. Dozenten:	<ul style="list-style-type: none"> • Stefan Zimmer • Michael Bader • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 080300100 Mathematik für Informatiker und Softwaretechniker • 051240005 Numerische und Stochastische Grundlagen der Informatik 		
12. Lernziele:	Beherrschung des grundsätzlichen Vorgehens in der Modellbildung. Kenntnis einer Auswahl diskreter und kontinuierlicher Modelle und entsprechender Simulationsmethoden. Fähigkeit, mit den erlernten Kenntnissen selbständig numerische Methoden problemorientiert um- und einzusetzen.		
13. Inhalt:	Diese Vorlesung bietet eine Einführung in die Grundlagen der Modellbildung und Simulation mit dem Ziel der Vorbereitung auf weiterführende Vorlesungen in diesem Bereich. Da Simulationsmethoden oft für viele verschiedene Problemklassen einsetzbar sind, ist die Vorlesung methodisch strukturiert. Den Hauptteil der Vorlesung bilden hierbei kontinuierliche Modelle sowie deren numerische Behandlung. Vorgestellt werden insbesondere die Diskretisierung gewöhnlicher und partieller Differentialgleichungen mit Finiten Differenzen sowie die Lösung großer dünnbesetzter Gleichungssysteme, Eigenwert- und Minimierungsprobleme. Diese Verfahren werden auf Problemstellungen der Natur- und Ingenieurwissenschaften angewendet, z.B. Populationwachstum, Wärmetransport und Verkehrssimulation.		
14. Literatur:	<ul style="list-style-type: none"> • Modellbildung und Simulation - Eine anwendungsorientierte Einführung; Bungartz, H.-J., Zimmer, S., Buchholz, M., Pflüger, D. , Springer Verlag, eXamen.press, 2009, ISBN 978-3-540-79809-5 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101201 Vorlesung Grundlagen der Modellbildung und Simulation • 101202 Übung Grundlagen der Modellbildung und Simulation 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	10121 Grundlagen der Modellbildung und Simulation (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 3. Semester
 - Ergänzungsmodule
 - Katalog ISG 1-3
- BA (Komb) Informatik, 4. Semester
 - Module im Nebenfach
 - Katalog ISG
- Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Ergänzendes Modul
- Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Wahlmodule

Modul: 10150 Grundlagen des Compilerbaus und der Programmiersprachen

2. Modulkürzel:	051510015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog SWT		
11. Voraussetzungen:	Grundkenntnisse aus den Einführungsvorlesungen des Informatikgrundstudiums, sowie einige Erfahrungen mit Programmierung. Vorkenntnisse über formale Sprachen sind vorteilhaft, aber nicht zwingend.		
12. Lernziele:	Die Studierenden haben die Grundkenntnisse erlangt, die zur effizienten Verwendung von Lexer- und Parser-Generatoren zur Analyse von Eingabetexten nötig sind. Sie haben gelernt, die Fehlermeldungen aus diesen Generatoren und den Compilern oder Interpretern richtig einzuordnen. Ferner haben sie durch Betrachtung der Implementierungsmodelle typischer Programmiersprachenkonstrukte Verständnis für das Ausführungsverhalten und für typische, gefährliche Fehlerquellen in Anwendungsprogrammen erlangt.		
13. Inhalt:	Compilerarchitekturen im Überblick; lexikalische und syntaktische Analyse von Texten mit formaler Grammatik, insb. von Programmiersprachen. Lexikalische Analyse: endliche Automaten und ihre Implementierung; Syntaxanalyse: diverse Parser-Strategien, ihre Implementierung und Eigenschaften. Methoden der automatischen Generierung von Analysatoren aus Spezifikationen der Grammatiken. Fehlererkennung und -behandlung. Analyse der statischen Semantik: Grundbegriffe und elementare Methoden. Eigenschaften von Programmiersprachen; Realisierung der Laufzeitsemantik prozeduraler Programmiersprachen aus Benutzersicht, insbesondere Implementierungsmodelle der Speicherverwaltung und der Unterprogrammaufrufe. Vermeidung typischer Fehlerquellen und überraschender Probleme in Anwendungsprogrammen.		
14. Literatur:	<ul style="list-style-type: none"> • Aho, Sethi, Ullman, Compilers - Principles, Techniques, and Tools, 1988 • Wilhelm, Maurer, Uebersetzerbau, 1997 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101501 Vorlesung Grundlagen des Compilerbaus und der Programmiersprachen • 101502 Übung Grundlagen des Compilerbaus und der Programmiersprachen 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	

Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name: 10151 Grundlagen des Compilerbaus und der Programmiersprachen (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 3. Semester
 - Ergänzungsmodule
 - Katalog ISG 1-3
- BA (Komb) Informatik, 5. Semester
 - Module im Nebenfach
 - Katalog ISG
- Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Ergänzendes Modul
- Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Wahlmodule

Modul: 10210 Mensch-Computer-Interaktion

2. Modulkürzel:	051900001	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Albrecht Schmidt		
9. Dozenten:	<ul style="list-style-type: none"> • Albrecht Schmidt • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051200005 Systemkonzepte und -programmierung 		
12. Lernziele:	<p>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.</p>		
13. Inhalt:	<p>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.</p> <p>Die folgenden Themen werden in der Vorlesung behandelt:</p> <ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none"> • Bernhard Preim, Raimund Dachsel. Interaktive Systeme 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung. Springer, Berlin; 2. Auflage. 2010 • Alan Dix, Janet Finley, Gregory Abowd, Russell Beale, Human-Computer Interaction, 2004 • Ben Shneiderman, Catherine Plaisant, Designing the User Interfaces, 2005
<hr/>	
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 102101 Vorlesung Mensch-Computer-Interaktion • 102102 Übung Mensch-Computer-Interaktion
<hr/>	
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
<hr/>	
17. Prüfungsnummer/n und -name:	10211 Mensch-Computer-Interaktion (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
<hr/>	
18. Grundlage für ... :	
<hr/>	
19. Medienform:	
<hr/>	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme
<hr/>	
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 4. Semester → Kernmodule B.Sc. Wirtschaftsinformatik, 6. Semester → Informatik (B 1) B.Sc. Wirtschaftsinformatik, 6. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich II: Informatik Basis B.Sc. Wirtschaftsinformatik, 6. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich III: Informatik Simulation B.Sc. Maschinelle Sprachverarbeitung, 5. Semester → Wahlbereich E/I

Modul: 10220 Modellierung

2. Modulkürzel:	052010001	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> • Bernhard Mitschang • Frank Leymann 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen • 051200005 Systemkonzepte und -programmierung 		
12. Lernziele:	Am Ende des Moduls sind die Studierenden in der Lage, wesentliche Artefakte eines IT Systems zu modellieren. Der Zusammenhang und das Zusammenspiel solcher Artefakte ist verstanden. Die Rolle von Metamodellen und deren Erstellung ist klar.		
13. Inhalt:	<ul style="list-style-type: none"> • Entity-Relationship Modell & komplexe Objekte • Relationenmodell & Relationenalgebra , Überblick SQL • Transformationen von ER nach Relationen, Normalisierung • XML, DTD, XML-Schema, Info-Set, Namensräume • Metamodelle & Repository • RDF, RDF-S & Ontologien • UML • Petri Netze, Workflownetze • BPMN 		
14. Literatur:	<ul style="list-style-type: none"> • A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002 • R. Eckstein, S. Eckstein, "XML und Datenmodellierung", dpunkt.verlag 2004 • M. Hitz, G. Kappel, E. Kapsammer, W. Retschitzegger, UML @ Work - Objektorientierte Modellierung mit UML2, 2005 • P. Hitzler, M. Krötzsch, S. Rudolph, Y. Sure, Semantic Web, 2008 • T.J. Teorey, Database Modeling & Design, 2nd Edition, 1994 • H.J. Habermann, F. Leymann, "Repository", Oldenbourg 1993 • W. Reisig, "Petri-Netze", Vieweg & Teubner 2010 • B. Silver, "BPMN Method & Style", Cody-Cassidy Press 2009 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 102201 Vorlesung Modellierung • 102202 Übung Modellierung 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	10221 Modellierung (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein		
18. Grundlage für ... :	<ul style="list-style-type: none"> • 10030 Architektur von Anwendungssystemen • 10080 Datenbanken und Informationssysteme 		
19. Medienform:			

20. Angeboten von:

-
21. Zuordnung zu weiteren Curricula:
- B.Sc. Informatik, 4. Semester
 - Kernmodule
 - B.Sc. Wirtschaftsinformatik, 4. Semester
 - Informatik (B 1)
 - B.Sc. Technikpädagogik, 4. Semester
 - Hauptfach Informatik
 - Pflichtmodule Informatik
 - M.Sc. Technikpädagogik, 4. Semester
 - Wahlpflichtfach B
 - Affines Wahlpflichtfach Informatik
 - Affines Wahlpflichtfach Informatik Pflichtmodule
-

Modul: 16790 Rechnerorganisation 1

2. Modulkürzel:	051700006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	-
8. Modulverantwortlicher:		Hans-Joachim Wunderlich	
9. Dozenten:		Hans-Joachim Wunderlich	
10. Zuordnung zum Curriculum:		B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog SWT	
11. Voraussetzungen:		<ul style="list-style-type: none"> • Technische Grundlagen der Informatik (10930) 	
12. Lernziele:		<ul style="list-style-type: none"> • Kenntnisse über die Grundlagen der Organisation von Rechnern und deren maschinennahe Programmierung • Grundzüge über die Beschreibung und den Entwurf von Hardwaresystemen 	
13. Inhalt:		<p>Aufbau, Entwurf und maschinennahe Programmierung von Mikroprozessoren werden erläutert. In den Übungen wird das Wissen durch schriftliche Aufgaben sowie Experimente mit Prozessorsimulatoren vertieft. Im einzelnen werden behandelt:</p> <ul style="list-style-type: none"> • Grundstrukturen: Stack-, Akkumulator- und Register-basierende Rechner • Informationsdarstellung in Hardware: Daten- und Befehlsformate, Fehlererkennung und -korrektur, Befehlssätze und Unterstützung von Hochsprachen. • MIPS als RISC-Bespiel und seine Assemblerprogrammierung • Grundelemente und Entwurf auf Register-Transfer-Ebene • Grundzüge einer Hardware-Beschreibungssprache (VHDL) • Operationswerke: Multiplikation, Division, Gleitkommaeinheiten • Steuerwerksentwurf und Mikroprogrammierung • Befehlszyklus und Unterbrechungen • Pipelining und statisches Scheduling • Speicherorganisation: Cachestrukturen und virtueller Speicher, Seitenverwaltung, Segmentierung, TLB, MMU und DMA • Leistungsbewertung: Maßzahlen und CPI, Benchmarking und einfache Anwendung von Warteschlangen 	
14. Literatur:		Literatur, siehe Webseite zur Veranstaltung	
15. Lehrveranstaltungen und -formen:		<ul style="list-style-type: none"> • 167901 Vorlesung Rechnerorganisation 1 • 167902 Praktische und theoretische Übungen Rechnerorganisation 1 	
16. Abschätzung Arbeitsaufwand:		Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden	

17. Prüfungsnummer/n und -name: • 16791 Rechnerorganisation 1 (PL), schriftlich oder mündlich, 90 Min.,
Gewichtung: 7.0, 1.0
• 16792 Rechnerorganisation 1 - Übungsaufgaben (LBP), schriftlich,
eventuell mündlich, Gewichtung: 3.0

18. Grundlage für ... : 10140 Advanced Processor Architecture

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

Modul: 10330 Systemkonzepte und -programmierung

2. Modulkürzel:	051200005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> • Kurt Rothermel • Frank Leymann 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051520005 Programmierung und Software-Entwicklung • Modul 051510005 Datenstrukturen und Algorithmen 		
12. Lernziele:	<ul style="list-style-type: none"> • 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:	<p>Grundlegende Systemstrukturen - und organisationen</p> <ul style="list-style-type: none"> • Multitaskingsystem • Multiprozessorsystem • Verteiltes System <p>Modellierung und Analyse nebenläufiger Programme</p> <ul style="list-style-type: none"> • Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm • Korrektheit- und Leitungskriterien <p>Betriebssystemkonzepte</p> <ul style="list-style-type: none"> • Organisation von Betriebssystemen • Prozesse und Threads • Eingabe/Ausgabe • Scheduling <p>Konzepte zur Synchronisation über gemeinsamen Speicher</p> <ul style="list-style-type: none"> • Synchronisationsprobleme und -lösungen • Synchronisationswerkzeuge: Semaphore, Monitor <p>Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer</p> <ul style="list-style-type: none"> • Taxonomie: Kommunikation und Synchronisation • Nachrichten als Kommunikationskonzept • Höhere Kommunikationskonzepte <p>Basisalgorithmen für Verteilte Systeme</p> <ul style="list-style-type: none"> • Erkennung globaler Eigenschaften • Schnappschussproblem 		

	<ul style="list-style-type: none">• Konsistenter globaler Zustand• Verteilte Terminierung Praktische nebenläufige Programmierung in Java <ul style="list-style-type: none">• Threads und Synchronisation• Socketschnittstelle• RMI Programmierung
14. Literatur:	<ul style="list-style-type: none">• Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 103301 Vorlesung Systemkonzepte und -programmierung• 103302 Übung Systemkonzepte und -programmierung
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 10331 Systemkonzepte und -programmierung (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 7.0• 10332 Systemkonzepte und -programmierung - Übungsschein (LBP), schriftlich, eventuell mündlich, Gewichtung: 3.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 3. Semester → Kernmodule B.Sc. Technikpädagogik, 3. Semester → Hauptfach Informatik → Pflichtmodule Informatik M.Sc. Technikpädagogik, 3. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Pflichtmodule

Modul: 11110 Verteilte Systeme

2. Modulkürzel:	051200015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Kurt Rothermel		
9. Dozenten:	Kurt Rothermel		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen • 051200005 Systemkonzepte und -Programmierung 		
12. Lernziele:	<ul style="list-style-type: none"> • Verstehen der grundsätzlichen Eigenschaften, Konzepte und Verfahren verteilter Systeme. • Kann existierende verteilte Anwendungen und Systemplattformen hinsichtlich ihrer Eigenschaften analysieren und verstehen. • Kann verteilte Anwendungen/Systemplattformen auf der Grundlage der erlernten Methoden realisieren. • Kann sich mit Experten anderer Fachdisziplinen über die Anwendung verteilter Systeme verständigen. 		
13. Inhalt:	<ul style="list-style-type: none"> • Einführung in die verteilten Systeme • Systemmodelle • Kommunikation: Nachrichten, Remote Procedure Call (RPC), Remote Method Invocation RMI) • Namensgebung: Generierung und Resolution • Zeit und Uhren in verteilten Systemen: Anwendungen, logische Uhren, physikalische Uhren, Uhrensynchronisation • Prozesssynchronisation: Wechselseitiger Ausschluß • Globaler Zustand: Konzepte, Snapshot Algorithmus, verteiltes Debugging • Transaktionsmanagement: Serialisierbarkeit, Sperrverfahren, 2-Phasen-Commit-Protokolle • Datenreplikation: Primary Copy, Consensus-Protokolle und andere Algorithmen • Sicherheit: Verfahren zur Geheimhaltung, Integrität, Authentifikation und Autorisierung • Broadcast-Algorithmen: Verarbeitungsmodell, Broadcast-Semantiken und -Algorithmen 		
14. Literatur:	Literatur, siehe Webseite zur Veranstaltung		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 111101 Vorlesung Verteilte Systeme • 111102 Übungen Verteilte Systeme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 h	
	Selbststudiumszeit / Nachbearbeitungszeit:	138 h	
	Gesamt:	180 h	
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 11111 Verteilte Systeme (PL), schriftliche Prüfung, 90 Min., Gewichtung: 7.0, • 11112 Verteilte Systeme - Übungen (LBP), schriftlich, eventuell mündlich, Gewichtung: 3.0 		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Parallele und Verteilte Systeme

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 3. Semester
 - Ergänzungsmodule
 - Katalog ISG 1-3
- BA (Komb) Informatik, 5. Semester
 - Module im Nebenfach
 - Katalog ISG
- B.Sc. Wirtschaftsinformatik, 5. Semester
 - Informatik (B 1)
- B.Sc. Wirtschaftsinformatik, 5. Semester
 - Wahlpflichtbereich (Bereich C)
 - Wahlbereich II: Informatik Basis
- B.Sc. Wirtschaftsinformatik, 5. Semester
 - Wahlpflichtbereich (Bereich C)
 - Wahlbereich III: Informatik Simulation

330 Katalog ISW

Zugeordnete Module:	10140	Advanced Processor Architecture
	10040	Bildsynthese
	10050	Bildverstehen
	10060	Computergraphik
	10090	Grundlagen der Eingebetteten Systeme
	10100	Grundlagen der Graphischen Ingenieursysteme
	10110	Grundlagen der Künstlichen Intelligenz
	10120	Grundlagen der Modellbildung und Simulation
	10150	Grundlagen des Compilerbaus und der Programmiersprachen
	14380	Hardware Verification and Quality Assessment
	10170	Imaging Science
	10180	Information Retrieval und Text Mining
	10210	Mensch-Computer-Interaktion
	10250	Parallele Systeme
	10300	Rechnernetze
	16790	Rechnerorganisation 1
	10330	Systemkonzepte und -programmierung
	11330	Visualisierung

Modul: 10140 Advanced Processor Architecture

2. Modulkürzel:	051700010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> • Hans-Joachim Wunderlich • Stefan Holst 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051700005 Rechnerorganisation 		
12. Lernziele:	Good understanding of the basic concepts used in modern CPUs and computing systems. Awareness of the challenges in modern processor design and the reasoning behind current and future design trends.		
13. Inhalt:	Classic topics in computer architecture as hardware/software interface are discussed as well as more advanced topics which include: <ul style="list-style-type: none"> • Technology basics: Design patterns, fabrication, yield, test and reliability, cost and quality, scaling. • Performance: Frequency and instructions per clock cycle, performance analysis and optimization. • Power dissipation: Analysis and optimization of power and performance, power and scaling. • Computer arithmetic: Efficient hardware for basic arithmetic, implementation of exponential, logarithm and trigonometric functions, floating point arithmetic and standards, arithmetic pipelines and filter, real-world floating point implementations like the Cell SPE or SPARC. • Instruction parallelism: Super scalar computing, static and dynamic scheduling, out-of-order execution, VLIW-processors, multithreading • Parallel architectures: Shared memory and message passing, multi-core processors, multi-core systems on a chip and emerging many-core technologies found in current graphic accelerators • Memory hierarchy: Memory technology and cache design. • Fault tolerance for single processors and multi processor systems 		
14. Literatur:	<ul style="list-style-type: none"> • I. Koren, Computer Arithmetic Algorithms, 2001 • J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 2006 • S. Iman, M. Pedram, Logic Synthesis for Low Power VLSI Designs, 1997 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101401 Vorlesung Grundlagen der Rechnerarchitektur 		

• 101402 Übung Grundlagen der Rechnerarchitektur

16. Abschätzung Arbeitsaufwand: Präsenzzeit: 42 Stunden
Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name: 10141 Advanced Processor Architecture (PL), schriftliche Prüfung,
90 Min., Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula: B.Sc. Informatik, 3. Semester
→ Ergänzungsmodule
→ Katalog ISG 1-3
BA (Komb) Informatik, 5. Semester
→ Module im Nebenfach
→ Katalog ISG
Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
→ Ergänzendes Modul
Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
→ Wahlmodule

Modul: 10040 Bildsynthese

2. Modulkürzel:	051900012	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> • Martin Fuchs • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051900002 Computergraphik • Modul 051900001 Mensch-Computer-Interaktion 		
12. Lernziele:	<p>Die Studierenden besitzen Wissen über verschiedene Ansätze und Algorithmen der dreidimensionalen Computergraphik, physikalisch-basierte Verfahren wie Raytracing und Radiosity, die den Lichttransport und die Wechselwirkung mit Materie modellieren, und numerische Methoden wie Monte-Carlo-Integration und Finite-Elemente-Verfahren die es erlauben, die Rendering-Gleichung zu lösen. Interaktive Verfahren nutzen spezielle Eigenschaften moderner Graphikhardware, um mit Hilfe mehrdimensionaler Texturen und anderer Rasterisierungsoperationen realistische Beleuchtungseffekte in Echtzeit zu generieren. Bild-basierte Verfahren verzichten auf eine geometrische Repräsentation der Szene und erzeugen neue Ansichten aus anderen aufgenommenen Bildern.</p>		
13. Inhalt:	<p>In dieser Vorlesung werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none"> • Grafik Hardware und APIs, OpenGL • Texturen, prozedurale Modelle • Schattenberechnungen • Szenengraphen, Culling, Level-of-Detail Verfahren • Physikalisch-basierte Beleuchtungsberechnung, Fotorealistische Bildsynthese • Lokale Beleuchtungsmodelle • Raytracing, Monte-Carlo Methoden • Radiosity 		
14. Literatur:	<ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none">• 100401 Vorlesung Bildsynthese• 100402 Übung Bildsynthese
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10041 Bildsynthese (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Wahlmodule aus Master Informatik BA (Komb) Informatik, 4. Semester → Module im Nebenfach → Katalog ISW

Modul: 10050 Bildverstehen

2. Modulkürzel:	051200035	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Paul Levi		
9. Dozenten:	<ul style="list-style-type: none"> • Paul Levi • Viktor Avrutin 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	Grundwissen über Programmierung, Datenstrukturen und Mathematik.		
12. Lernziele:	<p>Die Teilnehmer kennen und verstehen die Grundlagen der klassischen, verkörperten und verteilten Künstlichen Intelligenz. Ihnen wurden die dabei verwendeten Grundbegriffe so vorgestellt, dass sie als Bausteine von Architekturen intelligenter Systeme betrachtet werden. Dies bedeutet im Einzelnen, dass die Zuhörer am Beispiel des Bildverstehens die Wirkungsweise kognitiver Wahrnehmungsfähigkeiten kennengelernt haben, verstehen wie diese Fähigkeiten im Rahmen des Beobachtungs-, Planungs- und Aktionszyklus in einem Agenten eingesetzt werden und wie diese Fähigkeiten erweitert werden müssen, damit einzelne Agenten sich in einem Team kooperativ (Multiagenten-Systeme) verhalten können.</p> <p>Die Studierenden sind in der Lage, nicht nur einzelne wesentliche Algorithmen der Künstlichen Intelligenz (einschließlich des Bildverstehens) zu verstehen, sondern auch die wechselseitigen Beziehungen von verschiedenen Methoden zu berücksichtigen, um eine Beurteilung der Konzeption und der Wirkungsweise von intelligenten (kognitiven) technischen Systemen selbst vornehmen zu können.</p>		
13. Inhalt:	<ul style="list-style-type: none"> • Einleitung in Problemstellungen und Definitionen • Einführung in die Grundlagen der Bildverarbeitung • (Künstliche) Neuronale Netze • Bedingungsausbreitung (Constraints und ihre Propagierung) • Probabilistische Inferenz mit Bayes-Netzwerke • Verteilte Künstliche Intelligenz (VKI) • Multiagentensysteme (MAS) 		
14. Literatur:	<ul style="list-style-type: none"> • G. F. Luger, Künstliche Intelligenz, 2001 • Jähne, Bernd, Digitale Bildverarbeitung, 2005 • Literatur, siehe Webseite zur Veranstaltung • S. Russell, P. Norvig, Künstliche Intelligenz, 2004 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 100501 Vorlesung Bildverstehen • 100502 Übung Bildverstehen 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	10051 Bildverstehen (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 4. Semester
 - Ergänzungsmodule
 - Katalog ISW 1-3
- BA (Komb) Informatik, 4. Semester
 - Module im Nebenfach
 - Katalog ISW
- B.Sc. Technische Kybernetik, 4. Semester
 - Ergänzungsmodule
 - Grundlagen der Natur- und Ingenieurwissenschaften
- B.Sc. Technische Kybernetik, 4. Semester
 - Ergänzungsmodule
 - Höhere Informatik

Modul: 10060 Computergraphik

2. Modulkürzel:	051900002	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051900001 Mensch-Computer-Interaktion • Modul 051240005 Numerik und Stochastik. 		
12. Lernziele:	Die Studierenden haben Wissen über die Grundlagen der Computergraphik sowie praktische Fähigkeiten in der Graphikprogrammierung erworben.		
13. Inhalt:	Folgende Themen werden in der Vorlesung behandelt: <ul style="list-style-type: none"> • Überblick über den Bildsyntheseprozess • Grundlegende Rastergraphik • 2D und 3D Geometrietransformationen, 3D Projektion • Polygonale und hierarchische Modelle • Verdeckungsrechnung • Grundlegende Renderingtechniken (Rasterung, Raytracing) • Beleuchtungsmodelle • Texturen • Grundlagen der geometrischen Modellierung (Kurven, Flächen) <p>Die Veranstaltung besteht aus Vorlesung mit Übungen. Die Übungen umfassen praktische Programmierübungen, theoretische Themen und Programmierprojekte.</p>		
14. Literatur:	<ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none"> • 100601 Vorlesung Computergraphik • 100602 Übung Computergraphik 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	10061 Computergraphik (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein.		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule		

→ Katalog ISW 1-3

BA (Komb) Informatik, 5. Semester

→ Module im Nebenfach

→ Katalog ISW

B.Sc. Maschinelle Sprachverarbeitung, 5. Semester

→ Wahlbereich E/I

Modul: 10090 Grundlagen der Eingebetteten Systeme

2. Modulkürzel:	051711010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051700005 Rechnerorganisation 		
12. Lernziele:	Methodische und werkzeugunterstützte Konstruktion von Hardware und Software für auf spezifische Anwendungen hin optimierte eingebettete Systeme.		
13. Inhalt:	Modelle zur Systemspezifikation, Modellierungssprachen und ihre Simulation, System- und Architektursynthese, Allokation von Ressourcen und Bindung von Aufgaben/Operationen, Hardware-Software-Partitionierung, Verfahren zur Ablaufplanung für parallele Architekturen, Optimierungsverfahren, Anwendungsspezifische Prozessoren, On-Chip/Board-Verbindungsnetzwerke.		
14. Literatur:	<ul style="list-style-type: none"> • J. Teich, Digitale Hardware/Software-Systeme, 2. Auflage, 2007 • P. Marwedel, Embedded System Design, 2006 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 100901 Vorlesung Grundlagen der Eingebetteten Systeme • 100902 Übung Grundlagen der Eingebetteten Systeme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 10091 Grundlagen der Eingebetteten Systeme (PL), schriftlich, eventuell mündlich, Gewichtung: 75.0 • 10092 Grundlagen der Eingebetteten Systeme - Rechnerübungen (LBP), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 25.0 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG		

Modul: 10100 Grundlagen der Graphischen Ingenieursysteme

2. Modulkürzel:	051400005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dieter Roller		
9. Dozenten:	Dieter Roller		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	Grundstudium		
12. Lernziele:	<ul style="list-style-type: none"> • Kenntnis und Verständnis von Modellen bei der Produktentwicklung • Grundkenntnisse über die wichtigsten Modellarten, Algorithmen und Datenstrukturen und Techniken für den Datenaustausch 		
13. Inhalt:	Inhalte: <ul style="list-style-type: none"> • Anforderungen an CAD-Systeme • zweidimensionale Modelle • dreidimensionale Modelle • interaktive Modellerstellung • Einführung in die Modifikationstechnik u. parametrische Modellierung • Methoden zur Modellmodifikation • Grundlagen der parametrischen Modellierung • Ansätze und Verfahren zur parametrischen Variantenerzeugung • Ausgewählte Anwendungsbeispiele • Überblick über weitergehende Modellieransätze • Datenverwaltung in CAD 		
14. Literatur:	<ul style="list-style-type: none"> • D. Roller, CAD - Effiziente Anpassungs- und Variantenkonstruktion, Springer-Verlag • Literatur, siehe Webseite zur Veranstaltung 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101001 Vorlesung Grundlagen der Graphischen Ingenieursysteme • 101002 Übung Grundlagen der Graphischen Ingenieursysteme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	
	Nachbearbeitungszeit:	138 Stunden	
17. Prüfungsnummer/n und -name:	10101 Grundlagen der Graphischen Ingenieursysteme (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, 4. Semester → Module im Nebenfach		

- Katalog ISG
 - Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Ergänzendes Modul
 - Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Wahlmodule
-

Modul: 10110 Grundlagen der Künstlichen Intelligenz

2. Modulkürzel:	051900205	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Gunther Heidemann		
9. Dozenten:	Gunther Heidemann		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 080300100 Mathematik für Informatiker und Softwaretechniker 		
12. Lernziele:	Der Student / die Studentin beherrscht die Grundlagen der Künstlichen Intelligenz, kann Probleme der KI selbständig einordnen und mit den erlernten Methoden und Algorithmen bearbeiten.		
13. Inhalt:	<ul style="list-style-type: none"> • Intelligenz • Agentenbegriff • Problemlösen durch Suchen, Suchverfahren • Probleme mit Rand- und Nebenbedingungen • Spiele • Aussagen- und Prädikatenlogik • Logikbasierte Agenten, Wissensrepräsentation • Inferenz • Planen • Unsicherheit, probabilistisches Schließen • Probabilistisches Schließen über die Zeit • Sprachverarbeitung • Entscheidungstheorie • Lernen 		
14. Literatur:	<ul style="list-style-type: none"> • G. F. Luger, Künstliche Intelligenz, 2001 • S. Russell, P. Norvig, Künstliche Intelligenz, 2004 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101101 Vorlesung Grundlagen der Künstlichen Intelligenz • 101102 Übung Grundlagen der Künstlichen Intelligenz 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	10111 Grundlagen der Künstlichen Intelligenz (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3		

- BA (Komb) Informatik, 5. Semester
 - Module im Nebenfach
 - Katalog ISG
 - B.Sc. Wirtschaftsinformatik, 5. Semester
 - Informatik (B 1)
 - B.Sc. Wirtschaftsinformatik, 5. Semester
 - Wahlpflichtbereich (Bereich C)
 - Wahlbereich II: Informatik Basis
 - B.Sc. Wirtschaftsinformatik, 5. Semester
 - Wahlpflichtbereich (Bereich C)
 - Wahlbereich III: Informatik Simulation
 - B.Sc. Maschinelle Sprachverarbeitung, 5. Semester
 - Wahlbereich E/I
-

Modul: 10120 Grundlagen der Modellbildung und Simulation

2. Modulkürzel:	051240010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Marc Alexander Schweitzer		
9. Dozenten:	<ul style="list-style-type: none"> • Stefan Zimmer • Michael Bader • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 080300100 Mathematik für Informatiker und Softwaretechniker • 051240005 Numerische und Stochastische Grundlagen der Informatik 		
12. Lernziele:	Beherrschung des grundsätzlichen Vorgehens in der Modellbildung. Kenntnis einer Auswahl diskreter und kontinuierlicher Modelle und entsprechender Simulationsmethoden. Fähigkeit, mit den erlernten Kenntnissen selbständig numerische Methoden problemorientiert um- und einzusetzen.		
13. Inhalt:	Diese Vorlesung bietet eine Einführung in die Grundlagen der Modellbildung und Simulation mit dem Ziel der Vorbereitung auf weiterführende Vorlesungen in diesem Bereich. Da Simulationsmethoden oft für viele verschiedene Problemklassen einsetzbar sind, ist die Vorlesung methodisch strukturiert. Den Hauptteil der Vorlesung bilden hierbei kontinuierliche Modelle sowie deren numerische Behandlung. Vorgestellt werden insbesondere die Diskretisierung gewöhnlicher und partieller Differentialgleichungen mit Finiten Differenzen sowie die Lösung großer dünnbesetzter Gleichungssysteme, Eigenwert- und Minimierungsprobleme. Diese Verfahren werden auf Problemstellungen der Natur- und Ingenieurwissenschaften angewendet, z.B. Populationwachstum, Wärmetransport und Verkehrssimulation.		
14. Literatur:	<ul style="list-style-type: none"> • Modellbildung und Simulation - Eine anwendungsorientierte Einführung; Bungartz, H.-J., Zimmer, S., Buchholz, M., Pflüger, D. , Springer Verlag, eXamen.press, 2009, ISBN 978-3-540-79809-5 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101201 Vorlesung Grundlagen der Modellbildung und Simulation • 101202 Übung Grundlagen der Modellbildung und Simulation 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	10121 Grundlagen der Modellbildung und Simulation (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 3. Semester
 - Ergänzungsmodule
 - Katalog ISG 1-3
- BA (Komb) Informatik, 4. Semester
 - Module im Nebenfach
 - Katalog ISG
- Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Ergänzendes Modul
- Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Wahlmodule

Modul: 10150 Grundlagen des Compilerbaus und der Programmiersprachen

2. Modulkürzel:	051510015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog SWT		
11. Voraussetzungen:	Grundkenntnisse aus den Einführungsvorlesungen des Informatikgrundstudiums, sowie einige Erfahrungen mit Programmierung. Vorkenntnisse über formale Sprachen sind vorteilhaft, aber nicht zwingend.		
12. Lernziele:	Die Studierenden haben die Grundkenntnisse erlangt, die zur effizienten Verwendung von Lexer- und Parser-Generatoren zur Analyse von Eingabetexten nötig sind. Sie haben gelernt, die Fehlermeldungen aus diesen Generatoren und den Compilern oder Interpretern richtig einzuordnen. Ferner haben sie durch Betrachtung der Implementierungsmodelle typischer Programmiersprachenkonstrukte Verständnis für das Ausführungsverhalten und für typische, gefährliche Fehlerquellen in Anwendungsprogrammen erlangt.		
13. Inhalt:	Compilerarchitekturen im Überblick; lexikalische und syntaktische Analyse von Texten mit formaler Grammatik, insb. von Programmiersprachen. Lexikalische Analyse: endliche Automaten und ihre Implementierung; Syntaxanalyse: diverse Parser-Strategien, ihre Implementierung und Eigenschaften. Methoden der automatischen Generierung von Analysatoren aus Spezifikationen der Grammatiken. Fehlererkennung und -behandlung. Analyse der statischen Semantik: Grundbegriffe und elementare Methoden. Eigenschaften von Programmiersprachen; Realisierung der Laufzeitsemantik prozeduraler Programmiersprachen aus Benutzersicht, insbesondere Implementierungsmodelle der Speicherverwaltung und der Unterprogrammaufrufe. Vermeidung typischer Fehlerquellen und überraschender Probleme in Anwendungsprogrammen.		
14. Literatur:	<ul style="list-style-type: none"> • Aho, Sethi, Ullman, Compilers - Principles, Techniques, and Tools, 1988 • Wilhelm, Maurer, Uebersetzerbau, 1997 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101501 Vorlesung Grundlagen des Compilerbaus und der Programmiersprachen • 101502 Übung Grundlagen des Compilerbaus und der Programmiersprachen 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	

Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name: 10151 Grundlagen des Compilerbaus und der Programmiersprachen (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 3. Semester
 - Ergänzungsmodule
 - Katalog ISG 1-3
- BA (Komb) Informatik, 5. Semester
 - Module im Nebenfach
 - Katalog ISG
- Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Ergänzendes Modul
- Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Wahlmodule

Modul: 14380 Hardware Verification and Quality Assessment

2. Modulkürzel:	051700020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> • Hans-Joachim Wunderlich • Michael Kochte 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051700005 Rechnerorganisation • 051711005 Technische Grundlagen der Informatik 		
12. Lernziele:	<ul style="list-style-type: none"> • Basic knowledge of methodologies and algorithms of functional and formal verification, diagnosis, test and design for testability of integrated circuits • Application of tools for simulation, verification and test insertion 		
13. Inhalt:	<p>Complex integrated circuits and systems are hardly designed fault free at first go. Also during production defects and an imperfect yield have to be expected. The course deals with the basic techniques to find and locate faults and defects in the design and in the manufactured, integrated system. The discussed methods are applied with the help of commercial and academic tools in exercises and labs.</p> <p>The course comprises:</p> <ul style="list-style-type: none"> • Validation: Simulation and emulation in different design levels. • Formal verification: Equivalence checking and model checking. • Test: Fault simulation and test generation. • Debug and diagnosis. 		
14. Literatur:	<ul style="list-style-type: none"> • G. D. Hachtel, F. Somenzi, Logic Synthesis and Verification Algorithms, 2006 • K. L. McMillan, Symbolic Model Checking, 1993 • L.-T. Wang, C.-W. Wu, X. Wen, VLSI Test Principles and Architectures - Design for Testability, 2006 • M. L. Bushnell, V. D. Agrawal, Essentials of Electronic Testing, 2005 • R. Drechsler, B. Becker, Graphenbasierte Funktionsdarstellung, 2000 • S. Hassoun, T. Sasao, Logic Synthesis and Verification, 2002 • S. Minato, Binary Decision Diagrams and Applications for VLSI CAD, 1996 • T. Kropf, Introduction to Formal Hardware Verification, 1999 • W. Kunz, D. Stoffel, Reasoning in Boolean Networks, 1997 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 143801 Vorlesung Hardware Verification and Quality Assessment • 143802 Übung Hardware Verification and Quality Assessment 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	14381 Hardware Verification and Quality Assessment (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 4. Semester
 - Ergänzungsmodule
 - Wahlmodule aus Master Informatik
- BA (Komb) Informatik, 4. Semester
 - Module im Nebenfach
 - Katalog ISW

Modul: 10170 Imaging Science

2. Modulkürzel:	051900210	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Gunther Heidemann		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf • Gunther Heidemann 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 080300100 Mathematik für Informatiker und Softwaretechniker 		
12. Lernziele:	<p>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.</p> <p>The student knows the basics of digital image representation and processing and is able to solve problems of the field using the methods presented in the course.</p>		
13. Inhalt:	<ul style="list-style-type: none"> • Grundlagen aus der Optik:Lochkamera, Linsengleichung • Bildaufnahme:Kameras, Objektive, Beleuchtung, Aufnahmeprozess • Bildrepräsentation:Diskretisierung, Farbräume, Bildformate (z.B. ppm, gif, jpeg) • Elementare Bildbearbeitung:Punktoperationen (z.B. Kontrastverstärkung, Binarisierung) • Lineare und nichtlineare Filter:Faltung, morphologische Operatoren • Fouriertransformation, Bilddarstellung und -bearbeitung im Fourierraum, Abtasttheorem • Orthogonale Transformationen:Cosinus, Wavelets • Kompression:Generische Verfahren (RLE, Entropie), spezielle Bildverfahren (z.B. jpeg) • Video:Formate, Kompression (z.B. AVI, MPEG) • Bildverbesserung und Restauration • Elementare Segmentierungsverfahren:Histogramme, Farben, Konturen • Elementare Mustererkennung <ul style="list-style-type: none"> • Fundamentals of optics such as pinhole camera and lens equation • Image acquisition: Cameras, lenses, illumination, acquisition process • Image representation: Discretization, color spaces, image file formats (e.g. ppm, gif, jpeg) • Basics of image processing, e.g. point operations such as contrast enhancement or binarization • Linear and nonlinear filtering such as convolution and morphological operations. • Fourier transform, image representation and processing in Fourier space, sampling theorem • Orthogonal transforms such as cosine transform and wavelets • Compression: Generic compression (RLE, entropy coding), methods specialized to domain of images (e.g. jpeg) • Video: file formats, compression (e.g. avi, mpeg) 		

	<ul style="list-style-type: none"> • Image enhancement and restauration • Basics of segmentation: Histograms, colors, contours • Basics of pattern recognition
14. Literatur:	<ul style="list-style-type: none"> • Bässmann, Henning; Kreyss, Jutta, Bildverarbeitung Ad Oculos, 2004 • Forsyth, David and Ponce, Jean, Computer Vision. A Modern Approach.: A Modern Approach Computer Vision. A Modern Approach, 2003 • Gonzalez, Rafael C.; Woods, Richard E.; Eddins, Steven L., Digital Image Processing, 2004 • Jähne, Bernd, Digitale Bildverarbeitung / Digital Image Processing, 2005 • Klaus D. Tönnies, Grundlagen der Bildverarbeitung, 2005 • L. G. Shapiro, G. C. Stockman, Computer Vision, 2001
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101701 Vorlesung Imaging Science • 101702 Übung Imaging Science
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10171 Imaging Science (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG

Modul: 10180 Information Retrieval und Text Mining

2. Modulkürzel:	052401010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Hinrich Schütze		
9. Dozenten:	<ul style="list-style-type: none"> • Helmut Schmid • Hinrich Schütze 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	052400009		
12. Lernziele:	Die Studierenden haben ein grundlegendes Verständnis der Konzepte und Algorithmen des Information Retrieval und Text Mining entwickelt.		
13. Inhalt:	<ul style="list-style-type: none"> • Textpräprozessierung • invertierte Indexe • IR-Modelle (z.B. Vektorraum-basiertes IR) • Linkanalyse • Clustering • Frage-Antwort-Systeme • Informationsextraktion • korpusbasierter Erwerb von lexikalischem und Weltwissen 		
14. Literatur:	<ul style="list-style-type: none"> • Chris Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, 2008 Cambridge University Press 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101801 Vorlesung Information Retrieval and Text Mining • 101802 Übung Information Retrieval and Text Mining 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	10181 Information Retrieval und Text Mining (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 4. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISW B.Sc. Maschinelle Sprachverarbeitung, 5. Semester → Kernmodule		

Modul: 10210 Mensch-Computer-Interaktion

2. Modulkürzel:	051900001	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Albrecht Schmidt		
9. Dozenten:	<ul style="list-style-type: none"> • Albrecht Schmidt • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051200005 Systemkonzepte und -programmierung 		
12. Lernziele:	<p>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.</p>		
13. Inhalt:	<p>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.</p> <p>Die folgenden Themen werden in der Vorlesung behandelt:</p> <ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none">• Bernhard Preim, Raimund Dachsel. Interaktive Systeme 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung. Springer, Berlin; 2. Auflage. 2010• Alan Dix, Janet Finley, Gregory Abowd, Russell Beale, Human-Computer Interaction, 2004• Ben Shneiderman, Catherine Plaisant, Designing the User Interfaces, 2005
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 102101 Vorlesung Mensch-Computer-Interaktion• 102102 Übung Mensch-Computer-Interaktion
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10211 Mensch-Computer-Interaktion (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 4. Semester → Kernmodule B.Sc. Wirtschaftsinformatik, 6. Semester → Informatik (B 1) B.Sc. Wirtschaftsinformatik, 6. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich II: Informatik Basis B.Sc. Wirtschaftsinformatik, 6. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich III: Informatik Simulation B.Sc. Maschinelle Sprachverarbeitung, 5. Semester → Wahlbereich E/I

Modul: 10250 Parallele Systeme

2. Modulkürzel:	051200065	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 3. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	Ein Fach aus dem Bereich Technische Informatik.		
12. Lernziele:	Grundlegende Kenntnisse im Bereich paralleler Systeme, z.B. Multi-Core CPUs und deren Programmierung.		
13. Inhalt:	<ul style="list-style-type: none"> • Die Entwicklung vom klassischen Mikroprozessor zur Multi-Core CPU • Programmierung paralleler Rechnersysteme • Systolische Arrays, massiv parallele Systeme • Parallele Systeme aus verschiedenen • Anwendungsdomänen: ausgewählte Fallbeispiele 		
14. Literatur:	<ul style="list-style-type: none"> • Literatur, siehe Webseite zur Veranstaltung 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 102501 Vorlesung Parallele Systeme • 102502 Übung Parallele Systeme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	10251 Parallele Systeme (LBP), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 B.Sc. Informatik, 3. Semester → Ergänzungsmodule → Katalog ISW 1-3 BA (Komb) Informatik, 5. Semester → Module im Nebenfach → Katalog ISG B.Sc. Technische Kybernetik, 3. Semester → Ergänzungsmodule → Höhere Informatik		

Modul: 10300 Rechnernetze

2. Modulkürzel:	051200010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> • Kurt Rothermel • Frank Dürr 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051510005 Datenstrukturen und Algorithmen 		
12. Lernziele:	<ul style="list-style-type: none"> • Versteht grundsätzliche Eigenschaften, Konzepte und Methoden von Rechnernetzen, insbesondere dem Internet. • Versteht Schichten und deren Zusammenwirken in einem Protokollstapel • Kann Rechnernetze aufbauen, verwalten und analysieren. • Kann Protokolle entwickeln und in Schichtenarchitektur einbetten. • Kann höhere Kommunikationsdienste zur Entwicklung von netzgestützten Systemen anwenden. • Kann sich mit Experten anderer Domänen über Methoden der Rechnernetze verständigen. 		
13. Inhalt:	<ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none"> • 103001 VL Rechnernetze • 103002 ÜB Rechnernetze 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		

-
17. Prüfungsnummer/n und -name:
- 10301 Rechnernetze (PL), schriftlich, eventuell mündlich, 90 Min.,
Gewichtung: 7.0
 - 10302 Rechnernetze - Übungen (LBP), schriftlich, eventuell
mündlich, Gewichtung: 3.0
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:
- B.Sc. Informatik, 4. Semester
 - Ergänzungsmodule
 - Katalog ISW 1-3
 - BA (Komb) Informatik, 4. Semester
 - Module im Nebenfach
 - Katalog ISW
 - B.Sc. Technische Kybernetik, 4. Semester
 - Ergänzungsmodule
 - Höhere Informatik
-

Modul: 16790 Rechnerorganisation 1

2. Modulkürzel:	051700006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	-
8. Modulverantwortlicher:		Hans-Joachim Wunderlich	
9. Dozenten:		Hans-Joachim Wunderlich	
10. Zuordnung zum Curriculum:		B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog SWT	
11. Voraussetzungen:		<ul style="list-style-type: none"> • Technische Grundlagen der Informatik (10930) 	
12. Lernziele:		<ul style="list-style-type: none"> • Kenntnisse über die Grundlagen der Organisation von Rechnern und deren maschinennahe Programmierung • Grundzüge über die Beschreibung und den Entwurf von Hardwaresystemen 	
13. Inhalt:		<p>Aufbau, Entwurf und maschinennahe Programmierung von Mikroprozessoren werden erläutert. In den Übungen wird das Wissen durch schriftliche Aufgaben sowie Experimente mit Prozessorsimulatoren vertieft. Im einzelnen werden behandelt:</p> <ul style="list-style-type: none"> • Grundstrukturen: Stack-, Akkumulator- und Register-basierende Rechner • Informationsdarstellung in Hardware: Daten- und Befehlsformate, Fehlererkennung und -korrektur, Befehlssätze und Unterstützung von Hochsprachen. • MIPS als RISC-Bespiel und seine Assemblerprogrammierung • Grundelemente und Entwurf auf Register-Transfer-Ebene • Grundzüge einer Hardware-Beschreibungssprache (VHDL) • Operationswerke: Multiplikation, Division, Gleitkommaeinheiten • Steuerwerksentwurf und Mikroprogrammierung • Befehlszyklus und Unterbrechungen • Pipelining und statisches Scheduling • Speicherorganisation: Cachestrukturen und virtueller Speicher, Seitenverwaltung, Segmentierung, TLB, MMU und DMA • Leistungsbewertung: Maßzahlen und CPI, Benchmarking und einfache Anwendung von Warteschlangen 	
14. Literatur:		Literatur, siehe Webseite zur Veranstaltung	
15. Lehrveranstaltungen und -formen:		<ul style="list-style-type: none"> • 167901 Vorlesung Rechnerorganisation 1 • 167902 Praktische und theoretische Übungen Rechnerorganisation 1 	
16. Abschätzung Arbeitsaufwand:		Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden	

17. Prüfungsnummer/n und -name: • 16791 Rechnerorganisation 1 (PL), schriftlich oder mündlich, 90 Min.,
Gewichtung: 7.0, 1.0
• 16792 Rechnerorganisation 1 - Übungsaufgaben (LBP), schriftlich,
eventuell mündlich, Gewichtung: 3.0

18. Grundlage für ... : 10140 Advanced Processor Architecture

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

Modul: 10330 Systemkonzepte und -programmierung

2. Modulkürzel:	051200005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> • Kurt Rothermel • Frank Leymann 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051520005 Programmierung und Software-Entwicklung • Modul 051510005 Datenstrukturen und Algorithmen 		
12. Lernziele:	<ul style="list-style-type: none"> • 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:	<p>Grundlegende Systemstrukturen - und organisationen</p> <ul style="list-style-type: none"> • Multitaskingsystem • Multiprozessorsystem • Verteiltes System <p>Modellierung und Analyse nebenläufiger Programme</p> <ul style="list-style-type: none"> • Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm • Korrektheit- und Leitungskriterien <p>Betriebssystemkonzepte</p> <ul style="list-style-type: none"> • Organisation von Betriebssystemen • Prozesse und Threads • Eingabe/Ausgabe • Scheduling <p>Konzepte zur Synchronisation über gemeinsamen Speicher</p> <ul style="list-style-type: none"> • Synchronisationsprobleme und -lösungen • Synchronisationswerkzeuge: Semaphore, Monitor <p>Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer</p> <ul style="list-style-type: none"> • Taxonomie: Kommunikation und Synchronisation • Nachrichten als Kommunikationskonzept • Höhere Kommunikationskonzepte <p>Basisalgorithmen für Verteilte Systeme</p> <ul style="list-style-type: none"> • Erkennung globaler Eigenschaften • Schnappschussproblem 		

	<ul style="list-style-type: none">• Konsistenter globaler Zustand• Verteilte Terminierung Praktische nebenläufige Programmierung in Java <ul style="list-style-type: none">• Threads und Synchronisation• Socketschnittstelle• RMI Programmierung
14. Literatur:	<ul style="list-style-type: none">• Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 103301 Vorlesung Systemkonzepte und -programmierung• 103302 Übung Systemkonzepte und -programmierung
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 10331 Systemkonzepte und -programmierung (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 7.0• 10332 Systemkonzepte und -programmierung - Übungsschein (LBP), schriftlich, eventuell mündlich, Gewichtung: 3.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 3. Semester → Kernmodule B.Sc. Technikpädagogik, 3. Semester → Hauptfach Informatik → Pflichtmodule Informatik M.Sc. Technikpädagogik, 3. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Pflichtmodule

Modul: 11330 Visualisierung

2. Modulkürzel:	051900011	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 6. Semester → Ergänzungsmodule → Katalog ISW		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051900002 Computergraphik • 051900001 Mensch-Computer-Interaktion • 051240005 Numerik und Stochastik. 		
12. Lernziele:	Die Studierenden haben Wissen über Grundlagen, Algorithmen und Datenstrukturen für die Visualisierung sowie praktische Fähigkeiten durch die Arbeit mit Visualisierungssoftware erworben.		
13. Inhalt:	<p>Visualisierung behandelt alle Aspekte, die mit der visuellen Repräsentation von Daten aus wissenschaftlichen Experimenten, Simulationen, medizinischen Scannern, Datenbanken oder ähnlichen Datenquellen gewonnen werden, um zu einem tieferen Verständnis zu gelangen oder eine einfachere Darstellung komplexer Phänomene oder Sachverhalte zu erhalten. Um dieses Ziel zu erreichen, werden zum einen wohlbekannte Techniken aus dem Gebiet der interaktiven Computergraphik, zum anderen auch neu entwickelte Techniken angewendet.</p> <p>Entsprechend werden in dieser Vorlesung folgenden Themen behandelt:</p> <ul style="list-style-type: none"> • Einführung, Historie, Visualisierungspipeline • Datenakquise und -repräsentation (Abtasten, Rekonstruktion, Gitter, Datenstrukturen) • Wahrnehmungsaspekte • Grundlegende Konzepte visueller Abbildungen • Visualisierung von Skalarfeldern (Isoflächenextraktion, Volumenrendering) • Visualisierung von Vektorfelder (Teilchenverfolgung, texturbasierte Methoden, Topologie) • Tensorfelder, Multiattributdaten • Hochdimensionale Daten und Informationsvisualisierung 		
14. Literatur:	<ul style="list-style-type: none"> • C. D. Hansen, C. R. Johnson, The Visualization Handbook, 2005 • C. Ware, Information Visualization: Perception for Design, 2004 • H. Schumann, W. Müller, Visualisierung: Grundlagen und allgemeine Methoden, 2000 • K. Engel, M. Hadwiger, J. M. Kniss, C. Rezk-Salama, D. Weiskopf, Real-time Volume Graphics, 2006 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 113301 Vorlesung Visualisierung • 113302 Übungen Visualisierung 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		

17. Prüfungsnummer/n und -name: 11331 Visualisierung (PL), mündliche Prüfung, 30 Min., Gewichtung:
1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 4. Semester
 - Ergänzungsmodule
 - Wahlmodule aus Master Informatik
- BA (Komb) Informatik, 6. Semester
 - Module im Nebenfach
 - Katalog ISW
- B.Sc. Maschinelle Sprachverarbeitung, 5. Semester
 - Wahlbereich E/I

310 Katalog SWT

Zugeordnete Module: 10120 Grundlagen der Modellbildung und Simulation
 10150 Grundlagen des Compilerbaus und der Programmiersprachen
 10210 Mensch-Computer-Interaktion
 16790 Rechnerorganisation 1
 10330 Systemkonzepte und -programmierung

Inhalt der Module aus Studiengängen, an denen das Institut für Linguistik beteiligt ist (siehe Anlage des Modulhandbuchs)

360h

alle Basismodule sowie Kernmodul 1

Wahlpflichtmodul im 5. und 6. Fachsemester

die Lernziele sind der gewählte Moduls aus der Anlage

Modul: 10120 Grundlagen der Modellbildung und Simulation

2. Modulkürzel:	051240010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Marc Alexander Schweitzer		
9. Dozenten:	<ul style="list-style-type: none"> • Stefan Zimmer • Michael Bader • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 080300100 Mathematik für Informatiker und Softwaretechniker • 051240005 Numerische und Stochastische Grundlagen der Informatik 		
12. Lernziele:	Beherrschung des grundsätzlichen Vorgehens in der Modellbildung. Kenntnis einer Auswahl diskreter und kontinuierlicher Modelle und entsprechender Simulationsmethoden. Fähigkeit, mit den erlernten Kenntnissen selbständig numerische Methoden problemorientiert um- und einzusetzen.		
13. Inhalt:	Diese Vorlesung bietet eine Einführung in die Grundlagen der Modellbildung und Simulation mit dem Ziel der Vorbereitung auf weiterführende Vorlesungen in diesem Bereich. Da Simulationsmethoden oft für viele verschiedene Problemklassen einsetzbar sind, ist die Vorlesung methodisch strukturiert. Den Hauptteil der Vorlesung bilden hierbei kontinuierliche Modelle sowie deren numerische Behandlung. Vorgestellt werden insbesondere die Diskretisierung gewöhnlicher und partieller Differentialgleichungen mit Finiten Differenzen sowie die Lösung großer dünnbesetzter Gleichungssysteme, Eigenwert- und Minimierungsprobleme. Diese Verfahren werden auf Problemstellungen der Natur- und Ingenieurwissenschaften angewendet, z.B. Populationwachstum, Wärmetransport und Verkehrssimulation.		
14. Literatur:	<ul style="list-style-type: none"> • Modellbildung und Simulation - Eine anwendungsorientierte Einführung; Bungartz, H.-J., Zimmer, S., Buchholz, M., Pflüger, D. , Springer Verlag, eXamen.press, 2009, ISBN 978-3-540-79809-5 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101201 Vorlesung Grundlagen der Modellbildung und Simulation • 101202 Übung Grundlagen der Modellbildung und Simulation 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	10121 Grundlagen der Modellbildung und Simulation (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 1.0		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 3. Semester
 - Ergänzungsmodule
 - Katalog ISG 1-3
- BA (Komb) Informatik, 4. Semester
 - Module im Nebenfach
 - Katalog ISG
- Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Ergänzendes Modul
- Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Wahlmodule

Modul: 10150 Grundlagen des Compilerbaus und der Programmiersprachen

2. Modulkürzel:	051510015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog SWT		
11. Voraussetzungen:	Grundkenntnisse aus den Einführungsvorlesungen des Informatikgrundstudiums, sowie einige Erfahrungen mit Programmierung. Vorkenntnisse über formale Sprachen sind vorteilhaft, aber nicht zwingend.		
12. Lernziele:	Die Studierenden haben die Grundkenntnisse erlangt, die zur effizienten Verwendung von Lexer- und Parser-Generatoren zur Analyse von Eingabetexten nötig sind. Sie haben gelernt, die Fehlermeldungen aus diesen Generatoren und den Compilern oder Interpretern richtig einzuordnen. Ferner haben sie durch Betrachtung der Implementierungsmodelle typischer Programmiersprachenkonstrukte Verständnis für das Ausführungsverhalten und für typische, gefährliche Fehlerquellen in Anwendungsprogrammen erlangt.		
13. Inhalt:	Compilerarchitekturen im Überblick; lexikalische und syntaktische Analyse von Texten mit formaler Grammatik, insb. von Programmiersprachen. Lexikalische Analyse: endliche Automaten und ihre Implementierung; Syntaxanalyse: diverse Parser-Strategien, ihre Implementierung und Eigenschaften. Methoden der automatischen Generierung von Analysatoren aus Spezifikationen der Grammatiken. Fehlererkennung und -behandlung. Analyse der statischen Semantik: Grundbegriffe und elementare Methoden. Eigenschaften von Programmiersprachen; Realisierung der Laufzeitsemantik prozeduraler Programmiersprachen aus Benutzersicht, insbesondere Implementierungsmodelle der Speicherverwaltung und der Unterprogrammaufrufe. Vermeidung typischer Fehlerquellen und überraschender Probleme in Anwendungsprogrammen.		
14. Literatur:	<ul style="list-style-type: none"> • Aho, Sethi, Ullman, Compilers - Principles, Techniques, and Tools, 1988 • Wilhelm, Maurer, Uebersetzerbau, 1997 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101501 Vorlesung Grundlagen des Compilerbaus und der Programmiersprachen • 101502 Übung Grundlagen des Compilerbaus und der Programmiersprachen 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	

Nachbearbeitungszeit: 138 Stunden

17. Prüfungsnummer/n und -name: 10151 Grundlagen des Compilerbaus und der Programmiersprachen (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

- B.Sc. Informatik, 3. Semester
 - Ergänzungsmodule
 - Katalog ISG 1-3
- BA (Komb) Informatik, 5. Semester
 - Module im Nebenfach
 - Katalog ISG
- Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Ergänzendes Modul
- Allgemein Lehramt (GymPO I) ab PO 2010, 3. Semester
 - Wahlmodule

Modul: 10210 Mensch-Computer-Interaktion

2. Modulkürzel:	051900001	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Albrecht Schmidt		
9. Dozenten:	<ul style="list-style-type: none"> • Albrecht Schmidt • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, 4. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Voraussetzungen:	<ul style="list-style-type: none"> • 051520005 Programmierung und Software-Entwicklung • 051200005 Systemkonzepte und -programmierung 		
12. Lernziele:	<p>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.</p>		
13. Inhalt:	<p>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.</p> <p>Die folgenden Themen werden in der Vorlesung behandelt:</p> <ul style="list-style-type: none"> • 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:	<ul style="list-style-type: none"> • Bernhard Preim, Raimund Dachsel. Interaktive Systeme 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung. Springer, Berlin; 2. Auflage. 2010 • Alan Dix, Janet Finley, Gregory Abowd, Russell Beale, Human-Computer Interaction, 2004 • Ben Shneiderman, Catherine Plaisant, Designing the User Interfaces, 2005
<hr/>	
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 102101 Vorlesung Mensch-Computer-Interaktion • 102102 Übung Mensch-Computer-Interaktion
<hr/>	
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
<hr/>	
17. Prüfungsnummer/n und -name:	10211 Mensch-Computer-Interaktion (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein
<hr/>	
18. Grundlage für ... :	
<hr/>	
19. Medienform:	
<hr/>	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme
<hr/>	
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 4. Semester → Kernmodule B.Sc. Wirtschaftsinformatik, 6. Semester → Informatik (B 1) B.Sc. Wirtschaftsinformatik, 6. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich II: Informatik Basis B.Sc. Wirtschaftsinformatik, 6. Semester → Wahlpflichtbereich (Bereich C) → Wahlbereich III: Informatik Simulation B.Sc. Maschinelle Sprachverarbeitung, 5. Semester → Wahlbereich E/I

Modul: 16790 Rechnerorganisation 1

2. Modulkürzel:	051700006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	-
8. Modulverantwortlicher:	Hans-Joachim Wunderlich		
9. Dozenten:	Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog SWT		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Technische Grundlagen der Informatik (10930) 		
12. Lernziele:	<ul style="list-style-type: none"> • Kenntnisse über die Grundlagen der Organisation von Rechnern und deren maschinennahe Programmierung • Grundzüge über die Beschreibung und den Entwurf von Hardwaresystemen 		
13. Inhalt:	<p>Aufbau, Entwurf und maschinennahe Programmierung von Mikroprozessoren werden erläutert. In den Übungen wird das Wissen durch schriftliche Aufgaben sowie Experimente mit Prozessorsimulatoren vertieft. Im einzelnen werden behandelt:</p> <ul style="list-style-type: none"> • Grundstrukturen: Stack-, Akkumulator- und Register-basierende Rechner • Informationsdarstellung in Hardware: Daten- und Befehlsformate, Fehlererkennung und -korrektur, Befehlssätze und Unterstützung von Hochsprachen. • MIPS als RISC-Bespiel und seine Assemblerprogrammierung • Grundelemente und Entwurf auf Register-Transfer-Ebene • Grundzüge einer Hardware-Beschreibungssprache (VHDL) • Operationswerke: Multiplikation, Division, Gleitkommaeinheiten • Steuerwerksentwurf und Mikroprogrammierung • Befehlszyklus und Unterbrechungen • Pipelining und statisches Scheduling • Speicherorganisation: Cachestrukturen und virtueller Speicher, Seitenverwaltung, Segmentierung, TLB, MMU und DMA • Leistungsbewertung: Maßzahlen und CPI, Benchmarking und einfache Anwendung von Warteschlangen 		
14. Literatur:	Literatur, siehe Webseite zur Veranstaltung		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 167901 Vorlesung Rechnerorganisation 1 • 167902 Praktische und theoretische Übungen Rechnerorganisation 1 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		

17. Prüfungsnummer/n und -name: • 16791 Rechnerorganisation 1 (PL), schriftlich oder mündlich, 90 Min.,
Gewichtung: 7.0, 1.0
• 16792 Rechnerorganisation 1 - Übungsaufgaben (LBP), schriftlich,
eventuell mündlich, Gewichtung: 3.0

18. Grundlage für ... : 10140 Advanced Processor Architecture

19. Medienform:

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

Modul: 10330 Systemkonzepte und -programmierung

2. Modulkürzel:	051200005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> • Kurt Rothermel • Frank Leymann 		
10. Zuordnung zum Curriculum:	<p>B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISG</p> <p>B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, 5. Semester → Ergänzungsmodule → Katalog SWT</p>		
11. Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051520005 Programmierung und Software-Entwicklung • Modul 051510005 Datenstrukturen und Algorithmen 		
12. Lernziele:	<ul style="list-style-type: none"> • 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:	<p>Grundlegende Systemstrukturen - und organisationen</p> <ul style="list-style-type: none"> • Multitaskingsystem • Multiprozessorsystem • Verteiltes System <p>Modellierung und Analyse nebenläufiger Programme</p> <ul style="list-style-type: none"> • Abstraktionen: Atomare Befehle, Prozesse, nebenläufiges Programm • Korrektheit- und Leitungskriterien <p>Betriebssystemkonzepte</p> <ul style="list-style-type: none"> • Organisation von Betriebssystemen • Prozesse und Threads • Eingabe/Ausgabe • Scheduling <p>Konzepte zur Synchronisation über gemeinsamen Speicher</p> <ul style="list-style-type: none"> • Synchronisationsprobleme und -lösungen • Synchronisationswerkzeuge: Semaphore, Monitor <p>Konzepte zur Kommunikation und Synchronisation mittels Nachrichtentransfer</p> <ul style="list-style-type: none"> • Taxonomie: Kommunikation und Synchronisation • Nachrichten als Kommunikationskonzept • Höhere Kommunikationskonzepte <p>Basisalgorithmen für Verteilte Systeme</p> <ul style="list-style-type: none"> • Erkennung globaler Eigenschaften • Schnappschussproblem 		

	<ul style="list-style-type: none"> • Konsistenter globaler Zustand • Verteilte Terminierung Praktische nebenläufige Programmierung in Java <ul style="list-style-type: none"> • Threads und Synchronisation • Socketschnittstelle • RMI Programmierung
14. Literatur:	<ul style="list-style-type: none"> • Literatur, siehe Webseite zur Veranstaltung
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 103301 Vorlesung Systemkonzepte und -programmierung • 103302 Übung Systemkonzepte und -programmierung
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 10331 Systemkonzepte und -programmierung (PL), schriftlich, eventuell mündlich, 90 Min., Gewichtung: 7.0 • 10332 Systemkonzepte und -programmierung - Übungsschein (LBP), schriftlich, eventuell mündlich, Gewichtung: 3.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	
21. Zuordnung zu weiteren Curricula:	B.Sc. Informatik, 3. Semester → Kernmodule B.Sc. Technikpädagogik, 3. Semester → Hauptfach Informatik → Pflichtmodule Informatik M.Sc. Technikpädagogik, 3. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Pflichtmodule

400 Schlüsselqualifikationen fachaffin

Zugeordnete Module: 16610 Studienprojekt-Pr

Modul: 16610 Studienprojekt-Pr

2. Modulkürzel:	051520191	5. Moduldauer:	2 Semester
3. Leistungspunkte:	15.0 LP	6. Turnus:	unregelmäßig
4. SWS:	10.0	7. Sprache:	-
8. Modulverantwortlicher:	Jochen Ludewig		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum:	B.Sc. Softwaretechnik, 4. Semester → Schlüsselqualifikationen fachaffin		
11. Voraussetzungen:	Die Module Einführung in die Softwaretechnik, Programmentwicklung und Software-Praktikum müssen vor Beginn des Studienprojekts absolviert sein.		
12. Lernziele:	<p>Im Studienprojekt-Pr werden die Prinzipien der Kooperation in einem größeren, für die Praxis typischen Projekt angewendet und eingeübt. Dazu gehören die Kontakte zum Kunden (Anforderungsanalyse), die Projektplanung, die Kostenschätzung, die Qualitätssicherung und die Präsentation der Resultate in schriftlicher und mündlicher Form, auch die Techniken zur Konfliktlösung und zum Risiko-Management. Natürlich kommt auch das fachliche Wissen zur Realisierung eines Softwaresystems zum Zuge. Die Teilnehmer sind nach dem Projekt in der Lage, ein größeres Softwareprojekt zu organisieren und vollständig durchzuführen.</p>		
13. Inhalt:	<p>Die Teilnehmer entwickeln ein Softwaresystem nach Vorgaben des Kunden von der Angebotserstellung bis zur Übergabe. Störungen und Änderungen der Aufgabe im Projektverlauf sind normale Bestandteile des Projekts. Typisch beginnt das Studienprojekt mit der Erhebung der Anforderungen und der Anfertigung eines Angebots; darauf folgt die Entwicklung nach einem zu Beginn gewählten Prozessmodell. Das Projekt wird mit der Übergabe der Software in einer Präsentation abgeschlossen. Die Teilnehmer fertigen einen Bericht an, der die individuellen Leistungen erkennen lässt</p>		
14. Literatur:	-		
15. Lehrveranstaltungen und -formen:	166101 Praktikum Studienprojekt-Pr		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 210 Stunden Nachbearbeitungszeit: 210 Stunden		
17. Prüfungsnummer/n und -name:	16611 Studienprojekt-Pr (LBP), schriftlich oder mündlich, Gewichtung: 1.0, Ausreichende Leistungen im Studienprojekt-Pr werden in jedem der beiden Semester durch einen unbenoteten Schein bestätigt. Die Note im Studienprojekt-Pr wird auf der Grundlage der im Projekt gezeigten Leistungen und des Projektberichts, der die individuellen Beiträge der Teilnehmer angeben muss, bestimmt. Sie geht in die Gesamtnote des Studienprojekts mit dem Gewicht 5 ein; der andere Beitrag kommt von Studienprojekt-Th mit dem Gewicht 3.		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von:

21. Zuordnung zu weiteren Curricula:

700 Zusatzmodule

800 Schlüsselqualifikationen fachübergreifend gesamt

910 SQ Konto anerkannt
