



Universität Stuttgart

Modulhandbuch
Studiengang Master of Science Softwaretechnik
Prüfungsordnung: 2012

Wintersemester 2014/15
Stand: 30. September 2014

Universität Stuttgart
Keplerstr. 7
70174 Stuttgart

Kontaktpersonen:

Studiendekan/in:	Univ.-Prof. Stefan Wagner Institut für Softwaretechnologie Tel.: 0711/685-88455 E-Mail: stefan.wagner@informatik.uni-stuttgart.de
Studiengangsmanager/in:	Katrin Schneider Institut für Maschinelle Sprachverarbeitung Tel.: 685 88520 E-Mail: katrin.schneider@informatik.uni-stuttgart.de
Prüfungsausschussvorsitzende/r:	Univ.-Prof. Lars Grunske Zuverlässige Softwaresysteme Tel.: E-Mail: lars.grunske@iste.uni-stuttgart.de
Fachstudienberater/in:	Katrin Schneider Institut für Maschinelle Sprachverarbeitung Tel.: 685 88520 E-Mail: katrin.schneider@informatik.uni-stuttgart.de
Stundenplanverantwortliche/r:	Apl. Prof. Ulrich Hertrampf Institut für Formale Methoden der Informatik Tel.: 7816-344 E-Mail: ulrich.hertrampf@f05.uni-stuttgart.de

Inhaltsverzeichnis

Präambel	5
19 Auflagenmodule des Masters	6
11890 Algorithmen und Berechenbarkeit	7
16500 Software Engineering	8
100 Vertiefungsmodule	9
120 Vertiefungslinien	10
46450 Vertiefungslinie Architektur von Anwendungssystemen	11
29330 Vertiefungslinie Datenbanken und Informationssysteme	12
29340 Vertiefungslinie Intelligent Systems	14
46470 Vertiefungslinie Parallele Systeme	16
46440 Vertiefungslinie Programmiersprachen, Compilerbau, Programmanalysen	17
29370 Vertiefungslinie Rechnerarchitekturen und eingebettete Systeme	18
46460 Vertiefungslinie Sprachverarbeitung	19
29380 Vertiefungslinie Theoretische Informatik und Wissenschaftliches Rechnen	21
29390 Vertiefungslinie Verteilte Systeme	23
29400 Vertiefungslinie Visualisierung und Interaktive Systeme	25
110 Vertiefungsmodule Pflicht	27
42830 Entwicklungsprojekt	28
33350 Forschungsmethoden in der Softwaretechnik	29
42860 Hauptseminar (Master SWT 1)	30
55560 Hauptseminar (Master SWT 2)	32
42820 Prozessanalyse	33
36410 Requirements Engineering und Software-Architektur	34
42810 Software-Qualitätssicherung und -Wartung	35
200 Spezialisierungsmodule - MSWT	36
42910 Advanced Business Process Management	38
55600 Advanced Information Management	39
55740 Advanced Service Computing	40
29550 Algorithmische Geometrie	42
29560 Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems	43
42480 Ausgewählte Kapitel des Wissenschaftlichen Rechnens	45
56680 Automaten über unendlichen Objekten	47
10040 Bildsynthese	49
42900 Business Process Management	51
57050 Compilerbau	53
29570 Computer Interface Technologien	54
29430 Computer Vision	55
55640 Correspondence Problems in Computer Vision	57
29580 Data Compression	59
55620 Data Warehousing, Data Mining, and OLAP	60
11900 Design and Test of Systems-on-a-Chip	62
29600 Digital System Design II	64
29590 Digitale Systeme	65
39250 Distributed Systems I	66
45730 Distributed Systems II	68
29710 Embedded Systems Engineering	69
58190 Entwurf und Implementierung eines Compilers	70
56980 Fachpraktikum Cloud Architekturen und Management	71
29740 Fachpraktikum Eingebettete Systeme	72

24900 Fachpraktikum Rechnergestützte Methoden der Produktentwicklung	73
45750 Fachpraktikum Verteilte Systeme	75
29440 Geometric Modeling and Computer Animation	76
29450 Graphentheorie	78
29610 Hardware Based Fault Tolerance	79
42920 Hardware-Software-Codesign	81
42420 High Performance Computing	82
51720 IT-Strategy	84
48500 Image Synthesis	86
55630 Information Visualization and Visual Analytics	88
29630 Konzepte der Programmiersprachen	90
29460 Kryptographische Verfahren	92
29480 Loose Coupling and Message Based Applications	93
29470 Machine Learning	95
29640 Mikrocontroller	97
29720 Mobile Computing	99
29730 Modelling, Simulation, and Specification	101
55650 Multimodal Interaction for Ubiquitous Computers	103
40680 Optimization	104
56790 Parallele Numerik	106
29650 Parallele Programmierung	108
10250 Parallele Systeme	109
48570 Practical Course Visual Computing	110
29660 Programmanalysen und Compilerbau	111
51740 Quantencomputing	113
29670 Rapid Prototyping	114
29680 Real-Time Programming	115
29690 Real-Time Video Processing I	116
29700 Real-Time Video Processing II	117
45740 Rechnernetze II	118
48580 Reinforcement Learning	120
48600 Robotics I	122
48620 Scientific Visualization	123
29510 Service Computing	125
46660 Service Management and Cloud Computing, and Evaluation	127
42520 Services and Service Composition	128
56550 Software Verification	129
48640 Theoretical and Methodological Foundations of Autonomous Systems	131
46760 Theoretical and Methodological Foundations of Visual Computing	133
29500 Visual Computing	135
300 Fachaffine Schlüsselqualifikation	137
42850 Internetrecht	138
42840 Software-Recht	139
80620 Masterarbeit-SWT	140

Präambel

Der Master-Studiengang Softwaretechnik ist speziell auf die Absolventen des gleichnamigen Bachelorstudiengangs zugeschnitten, richtet sich aber auch an Informatiker mit Industrieerfahrung, die ihre Kenntnisse in den Methoden der Softwaretechnik vertiefen wollen. Er enthält daher diverse Pflichtmodule, welche fortgeschrittene Themen in diesem Bereich behandeln.

Der M.Sc. Softwaretechnik unterscheidet sich vom M.Sc. Informatik durch die Betonung der konstruktiven Aspekte in der Softwaretechnik. Durch eine Prozessanalyse und ein Entwicklungsprojekt bleiben Team-Arbeit und Praxis wichtig; hinzu kommen forschungsorientierte Schwerpunkte: "Forschungsmethoden in der Softwaretechnik", "Requirements Engineering und Architektur", "Qualitätssicherung und Wartung". Zu den Wahlfächern gehören Software-Recht und andere wichtige Randthemen. Zusätzlich wählen die Studierenden eine Vertiefungslinie, d.h. ein Fachgebiet der Informatik, mit dem sie sich besonders gründlich befassen. Den Abschluss des Studiums bildet eine Masterarbeit.

Die Studierenden des Masterstudiengangs Softwaretechnik können 30 Leistungspunkte, also ein volles Semester, individuell einsetzen, beispielsweise für spezielle Lehrveranstaltungen der Softwaretechnik oder der Informatik. Sie können sie aber auch für ein Semester an einer Universität im Ausland verwenden, dadurch wichtige Erfahrungen sammeln und ihren Horizont erweitern. Ein Auslandsstudium ist empfehlenswert und wird durch den organisatorischen Aufbau des Masterstudiengangs Softwaretechnik ermöglicht.

19 Auflagenmodule des Masters

Zugeordnete Module: 11890 Algorithmen und Berechenbarkeit
 16500 Software Engineering

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:	Univ.-Prof. Stefan Funke		
9. Dozenten:	<ul style="list-style-type: none"> • Stefan Funke • Volker Diekert • Ulrich Hertrampf 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 3. Semester → Kernmodule</p> <p>B.Sc. Softwaretechnik, PO 2012, 3. Semester → Kernmodule</p> <p>M.Sc. Softwaretechnik, PO 2012, 3. Semester → Auflagenmodule des Masters</p>		
11. Empfohlene 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 Entwurfsstrategien und Analysemethoden kennengelernt.		
13. Inhalt:	Berechenbarkeit vs. Unberechenbarkeit, Church'sche These, NP-Vollständigkeit, PSPACE-vollständige Algorithmen (QBF). Entwurfsstrategien: Teile und Herrsche, 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:	<ul style="list-style-type: none"> • 11891 Algorithmen und Berechenbarkeit (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min. 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Formale Methoden der Informatik		

Modul: 16500 Software Engineering

2. Modulkürzel:	51520170	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Lars Grunske		
9. Dozenten:	Lars Grunske		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 4. Semester → Kernmodule B.Sc. Softwaretechnik, PO 2012, 4. Semester → Kernmodule M.Sc. Softwaretechnik, PO 2012 → Auflagenmodule des Masters		
11. Empfohlene 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:	Ergänzend zur "Einführung in die Softwaretechnik" und daran anknüpfend behandelt diese Lehrveranstaltung folgende Themen: <ul style="list-style-type: none"> • Softwarequalitätssicherung • 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"> • Ludwig J., Lichter, H., Software Engineering - Grundlagen, Menschen, Prozesse, Techniken, 2. Aufl. 2010 • Liggesmeyer P., Software-Qualität. Testen, Analysieren und Verifizieren von Software. Spektrum Akademischer Verlag, 2002. 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 165001 Vorlesung Software Engineering • 165002 Übung Software Engineering • 256101 Vorlesung Grundlagen des Software Engineerings • 256102 Übung Grundlagen des Software Engineerings 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	16501 Software Engineering (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0,		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Softwaretechnologie		

100 Vertiefungsmodule

Zugeordnete Module:	110	Vertiefungsmodule Pflicht
	120	Vertiefungslinien

120 Vertiefungslinien

Zugeordnete Module:

- 29330 Vertiefungslinie Datenbanken und Informationssysteme
- 29340 Vertiefungslinie Intelligent Systems
- 29370 Vertiefungslinie Rechnerarchitekturen und eingebettete Systeme
- 29380 Vertiefungslinie Theoretische Informatik und Wissenschaftliches Rechnen
- 29390 Vertiefungslinie Verteilte Systeme
- 29400 Vertiefungslinie Visualisierung und Interaktive Systeme
- 46440 Vertiefungslinie Programmiersprachen, Compilerbau, Programmanalysen
- 46450 Vertiefungslinie Architektur von Anwendungssystemen
- 46460 Vertiefungslinie Sprachverarbeitung
- 46470 Vertiefungslinie Parallele Systeme

Modul: 46450 Vertiefungslinie Architektur von Anwendungssystemen

2. Modulkürzel:	[pord.modulcode]	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes Semester
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Frank Leymann		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungslinien		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Die Lernziele sind bei den einzelnen Modulen, die kombiniert werden, definiert.		
13. Inhalt:	Der Inhalt ist bei den einzelnen Modulen, die kombiniert werden, definiert.		
14. Literatur:	Literatur ist bei den einzelnen Modulen, die kombiniert werden, zu finden.		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 464501 Deutschsprachige Lehrveranstaltungen Wintersemester • 464502 Deutschsprachige Lehrveranstaltungen Sommersemester • 464503 courses in english - winter semester • 464504 courses in english - summer semester 		
16. Abschätzung Arbeitsaufwand:	abhängig von der gewählten Kombination ca: 2 x Präsenzzeit: 42 Stunden 2 x Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	46451 Vertiefungslinie Architektur von Anwendungssystemen (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Architektur von Anwendungssystemen		

Modul: 29330 Vertiefungslinie Datenbanken und Informationssysteme

2. Modulkürzel:	051210555	5. Moduldauer:	2 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes Semester
4. SWS:	8.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Bernhard Mitschang		
9. Dozenten:	<ul style="list-style-type: none"> • Bernhard Mitschang • Holger Schwarz 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungslinien		
11. Empfohlene Voraussetzungen:	Modellierungs-Vorlesung aus dem Bachelor oder gleichwertige Veranstaltungen		
12. Lernziele:	Die Studenten haben vertiefte Kenntnisse auf dem Gebiet der Informationssysteme erworben und können die erlernten Methoden erfolgreich zur Anfertigung wissenschaftlicher Arbeiten sowie zur Lösung von Problemen der Informatiosgewinnung, -verarbeitung und -verwaltung anwenden.		
13. Inhalt:	Es werden Vorlesungen mit Übungen im Umfang von 8 SWS besucht. Zum Vertiefungsmodul Informationssysteme gehören die Veranstaltungen: <ol style="list-style-type: none"> 1) Datenbanken und Informationssysteme (Vorlesung mit Übung, 4 SWS) 2) Advanced Information Management (Vorlesung mit Übung, 4 SWS) 3) Data Warehousing, Data Mining und OLAP-Technologien (Vorlesung mit Übung, 4 SWS) 4) Information Integration (Vorlesung mit Übung, 4 SWS) 		
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 Weitere Literatur wird in den einzelnen Lehrveranstaltungen bekanntgegeben.		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 293301 Deutschsprachige Lehrveranstaltungen Wintersemester • 293302 Deutschsprachige Lehrveranstaltungen Sommersemester • 293303 courses in english - winter semester • 293304 courses in english - summer semester 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 84 Stunden Selbststudium: 276 Stunden Gesamt: 360 Stunden		
17. Prüfungsnummer/n und -name:	29331 Vertiefungslinie Datenbanken und Informationssysteme (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von: Datenbanken und Informationssysteme

Modul: 29340 Vertiefungslinie Intelligent Systems

2. Modulkürzel:	051901555	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes Semester
4. SWS:	8.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Andrés Bruhn		
9. Dozenten:	<ul style="list-style-type: none"> • Andrés Bruhn • Marc Toussaint • Sebastian Pado 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungslinien</p>		
11. Empfohlene Voraussetzungen:	<p>Grundlegende Kenntnisse in Mathematik und Bildverarbeitung (z.B. Mathematik für Informatiker und Softwaretechniker 080300100, Imaging Science 051900210)</p> <p>Grundlagen der Künstlichen Intelligenz 10110</p>		
12. Lernziele:	<p>Die Studierenden haben vertiefte Kenntnisse in den Bereichen des Maschinensehens, des maschinellen Lernens, der maschinellen Sprachverarbeitung und der Robotik erworben und können mit den erlernten Methoden wissenschaftliche Arbeiten in diesem Bereich verstehen. Sie haben das notwendige Rüstzeug, um eine Masterarbeit in den zuvor genannten Gebieten anzufertigen.</p> <p>The students have acquired specialised knowledge in the areas of computer vision, machine learning, language processing, and robotics, and they are capable of understanding scientific papers and books from this field. They have the necessary knowledge to begin a Master's thesis in one of the aforementioned areas.</p>		
13. Inhalt:	<p>Es werden Vorlesungen bzw. Vorlesungen mit Übungen im Bereich Intelligente Systeme im Umfang von 8 SWS besucht, die im MINF-Katalog (MINF 1-8) den entsprechenden Verwendungshinweis tragen und dort inhaltlich beschrieben werden. In dem Vertiefungsmodul Intelligente Systeme gehören hierzu die Veranstaltungen:</p> <ol style="list-style-type: none"> 1) 051900215 Computer Vision (Vorlesung mit Übung, 4 SWS) 2) 051900211 Correspondence Problems in Computer Vision (Vorlesung mit Übung, 4 SWS) 3) 051900205 Grundlagen der künstlichen Intelligenz (Vorlesung mit Übung, 4 SWS) 4) 051200112 Maschine Learning (Vorlesung mit Übung, 4 SWS) 5) 051200113 Optimization (Vorlesung mit Übung, 4 SWS) 6) 051200888 Reinforcement Learning (Vorlesung mit Übung, 4 SWS) 		

- 7) 051200999 Robotics I: Introduction (Vorlesung mit Übung, 4 SWS)
- 8) N/A Advanced Robotics (Vorlesung mit Übung, 4 SWS)
- 9) 052401010 Information Retrieval und Text Mining (Vorlesung mit Übung, 4 SWS)

Students have to attend lectures with and without exercises in the field of Intelligent Systems with a total of 8 SWS. These lectures have to be eligible from the MINF-catalogue (MINF 1-8) where their content is described. The lectures in the specialised module Intelligent Systems include:

- 1) 051900215 Computer Vision (lecture with exercises, 4 SWS)
- 2) 051900211 Correspondence Problems in Computer Vision (lecture with exercises, 4 SWS)
- 3) 051900205 Grundlagen der künstlichen Intelligenz (lecture with exercises, 4 SWS)
- 4) 051200112 Maschine Learning (lecture with exercises, 4 SWS)
- 5) 051200113 Optimization (lecture with exercises, 4 SWS)
- 6) 051200888 Reinforcement Learning (lecture with exercises, 4 SWS)
- 7) 051200999 Robotics I: Introduction (lecture with exercises, 4 SWS)
- 8) N/A Advanced Robotics (lecture with exercises, 4 SWS)
- 9) 052401010 Information Retrieval und Text Mining (lecture with excercises, 4 SWS)

14. Literatur:

15. Lehrveranstaltungen und -formen:
- 293401 Deutschsprachige Lehrveranstaltungen Wintersemester
 - 293402 Deutschsprachige Lehrveranstaltungen Sommersemester
 - 293403 Courses in English - winter semester
 - 293404 Courses in English - summer semester

16. Abschätzung Arbeitsaufwand:
- Präsenzzeit: 84 Stunden
 Selbststudium: 276 Stunden
- Gesamt: 360 Stunden**

17. Prüfungsnummer/n und -name:
- 29341 Vertiefungslinie Intelligent Systems (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0
 - V Vorleistung (USL-V), schriftlich, eventuell mündlich

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 46470 Vertiefungslinie Parallele Systeme

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungslinien		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Die Studenten haben vertiefte Kenntnisse in den Bereichen parallele Systeme sowie in Multi-Core CPUs und deren Programmierung.		
13. Inhalt:	Es werden Vorlesungen bzw. Vorlesungen mit Übungen im Bereich Parallele Systeme im Umfang von (SWS besucht, die im MINF-Katalog (MINF 1-8) den entsprechenden Verwendungshinweis tragen und dort inhaltlich beschrieben werden. In dem Vertiefungsmodul Parallele Systeme gehören hierzu die Veranstaltungen: 1) Data Compression (Vorlesung mit Übung , 4SWS, 6LP) 2) Parallele Systeme (Vorlesung mit Übung, 4SWS , 6LP)		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 464701 Deutschsprachige Lehrveranstaltungen Wintersemester • 464702 Deutschsprachige Lehrveranstaltungen Sommersemester • 464703 courses in english - winter semester • 464704 courses in english - summer semester 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	84 Stunden	
	Selbststudium:	276 Stunden	
	Gesamt:	360 Stunden	
17. Prüfungsnummer/n und -name:	46471 Vertiefungslinie Parallele Systeme (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 46440 Vertiefungslinie Programmiersprachen, Compilerbau, Programmanalysen

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes Semester
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereeder		
9. Dozenten:			
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungslinien		
11. Empfohlene Voraussetzungen:			
12. Lernziele:			
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 464401 Deutschsprachige Lehrveranstaltungen Wintersemester • 464402 Deutschsprachige Lehrveranstaltungen Sommersemester • 464403 courses in english - winter semester • 464404 courses in english - summer semester 		
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	46441 Vertiefungslinie Programmiersprachen, Compilerbau, Programmanalysen (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Softwaretechnologie		

Modul: 29370 Vertiefungslinie Rechnerarchitekturen und eingebettete Systeme

2. Modulkürzel:	051700555	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes Semester
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> • Martin Radetzki • Hans-Joachim Wunderlich 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungslinien		
11. Empfohlene Voraussetzungen:	Grundlegende Kenntnisse in Rechnerarchitektur und eingebettete Systeme		
12. Lernziele:	Die Studierenden haben vertiefte Kenntnisse in mehreren Bereichen der Technischen Informatik, insbesondere der Rechnerarchitektur und der eingebetteten Systeme erworben und können mit den erlernten Methoden wissenschaftliche Arbeiten in diesem Bereich verstehen. Sie haben das notwendige Rüstzeug, um eine Masterarbeit im Gebiet der Technischen Informatik anzufertigen.		
13. Inhalt:	Es werden Vorlesungen bzw. Vorlesungen mit Übungen aus dem Bereich der Technischen Informatik im Umfang von 8 SWS besucht, die im MINF-Katalog (MINF 1-8) den entsprechenden Verwendungshinweis tragen und dort inhaltlich beschrieben werden		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 293701 Deutschsprachige Lehrveranstaltungen Wintersemester • 293702 Deutschsprachige Lehrveranstaltungen Sommersemester • 293703 courses in english - winter semester • 293704 courses in english - summer semester 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 84 Stunden Selbststudium: 276 Stunden		
17. Prüfungsnummer/n und -name:	29371 Vertiefungslinie Rechnerarchitekturen und eingebettete Systeme (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Technische Informatik		

Modul: 46460 Vertiefungslinie Sprachverarbeitung

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Sebastian Pado		
9. Dozenten:	Sebastian Pado		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungslinien		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Die Studierenden erhalten einen Überblick über die Grundkonzepte, Forschungsfragen, Ressourcen, Methoden und Anwendungen der Computerlinguistik. --- Students become familiar with the main concepts, research questions, resources, methods and applications of Computational Linguistics.		
13. Inhalt:	Dieses Modul setzt sich zusammen aus 1. dem Modul 35150 "Methods in Computational Linguistics" und 2. einem wählbaren Modul aus dem Katalog MCL 6 des Studiengangs M.Sc. Computational Linguistics. --- This module consists of 1. the module 35150 "Methods in Computational Linguistics" and 2. one module to be chosen from the catalogue MCL 6 of the program of study M.Sc. Computational Linguistics.		
14. Literatur:	Daniel Jurafsky and James H. Martin, 2008, Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Prentice Hall.		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 464601 Deutschsprachige Lehrveranstaltungen Wintersemester • 464602 Deutschsprachige Lehrveranstaltungen Sommersemester • 464603 courses in english - winter semester • 464604 courses in english - summer semester 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 112h, Selbststudium: 240h		
17. Prüfungsnummer/n und -name:	• 46461 Vertiefungslinie Sprachverarbeitung (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0		

-
- V Vorleistung (USL-V), schriftlich, eventuell mündlich
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 29380 Vertiefungslinie Theoretische Informatik und Wissenschaftliches Rechnen

2. Modulkürzel:	050420555	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes Semester
4. SWS:	8.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	<ul style="list-style-type: none"> • Volker Diekert • Stefan Funke • Ulrich Hertrampf • Miriam Mehl • Dirk Pflüger • Stefan Zimmer 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungslinien		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Die Studierenden kennen über das Wissen der Grundlagenvorlesungen hinaus detaillierte Methoden zur Lösung von Problemstellungen in zwei Teilgebieten der Theoretischen Informatik oder des Wissenschaftlichen Rechnens. Sie sind in der Lage, die Eignung von Methoden für eine gegebene Fragestellung zu beurteilen, die gelernten Verfahren geeignet anzuwenden und Vorschläge zur Modifikation von Verfahren zu machen, um neue Problemklassen zu bearbeiten.		
13. Inhalt:	Es werden Vorlesungen mit Übungen im Umfang von 8 SWS besucht. Zum Vertiefungsmodul Theoretische Informatik und Wissenschaftliches Rechnen gehören u.a. folgende Veranstaltungen mit jeweils 4 SWS (Vorlesung mit Übung): <ol style="list-style-type: none"> 1. Algorithmische Geometrie 2. Ausgewählte Kapitel des Wissenschaftlichen Rechnens 3. Automaten über unendlichen Objekten 4. Discrete Optimization (auf Englisch) 5. High Performance Computing (auf Englisch) 6. Konkrete Mathematik 7. Parallele Numerik 8. Quantencomputing 		
14. Literatur:	Die empfohlene Literatur wird bei den jeweiligen Lehrveranstaltungen angegeben.		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 293801 Deutschsprachige Lehrveranstaltungen Wintersemester • 293802 Deutschsprachige Lehrveranstaltungen Sommersemester • 293803 courses in english - winter semester 		

	<ul style="list-style-type: none">• 293804 courses in english - summer semester
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 84 Stunden Nachbearbeitungszeit: 276 Stunden Gesamtzeit: 360 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 29381 Vertiefungslinie Theoretische Informatik und Wissenschaftliches Rechnen (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0• V Vorleistung (USL-V), schriftlich, eventuell mündlich, 45 Min.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Formale Methoden der Informatik

Modul: 29390 Vertiefungslinie Verteilte Systeme

2. Modulkürzel:	051200555	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes Semester
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> • Kurt Rothermel • Frank Dürr • Boris Koldehofe • Muhammad Tariq 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungslinien		
11. Empfohlene Voraussetzungen:	Grundlegende Kenntnisse im Bereich der Systemkonzepte und -Programmierung (z.B. Kursnummer 021506) und der Rechnernetze (z.B. Kursnummer 021531).		
12. Lernziele:	Die Studierenden haben vertiefende Kenntnisse im Bereich der Verteilten Systeme, der Mobilten Systeme sowie der Rechnernetze erworben. Sie können mit den erlernten Methoden wissenschaftliche Arbeiten in diesen Bereichen verstehen und sind somit in der Lage, eine Masterarbeit in diesen Themengebieten anzufertigen. Außerdem haben sie das Rüstzeug, die in diesen Bereichen existierende Technologien zu analysieren, zu bewerten und weiterzuentwickeln. Durch die Teilnahme an Praktika und Programmierübungen werden sie vorbereitet, das fundierte Methodenwissen in der Praxis anzuwenden.		
13. Inhalt:	Es werden Vorlesungen bzw. Vorlesungen mit Übungen zu den Themengebieten Verteilte Systeme, Mobile Systeme sowie Rechnernetze im Umfang von 8SWS besucht, die im die im MINF-Katalog (MINF 1-8) den entsprechenden Verwendungshinweis tragen und dort inhaltlich beschrieben werden. In dem Vertiefungsmodul Verteilte Systeme gehören hierzu die Veranstaltungen: <ol style="list-style-type: none"> 1) Verteilte Systeme (Vorlesung mit Übung, 6 LP) 2) Mobile Computing (Vorlesung mit Übung, 6 LP) 3) Verteilte Algorithmen (Vorlesung, 3 LP) 4) Asynchronous Middleware Systems (Vorlesung, 3 LP) 5) Höhere Kommunikationskonzepte und -protokolle (Vorlesung, 3 LP) 6) Konzepte der Peer-to-Peer-Systeme (Vorlesung, 3 LP) 		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 293901 Deutschsprachige Lehrveranstaltungen Wintersemester • 293902 Deutschsprachige Lehrveranstaltungen Sommersemester • 293903 courses in english - winter semester • 293904 courses in english - summer semester 		
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 29391 Vertiefungslinie Verteilte Systeme (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 45 Min. 		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Verteilte Systeme

Modul: 29400 Vertiefungslinie Visualisierung und Interaktive Systeme

2. Modulkürzel:	051900555	5. Moduldauer:	2 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Martin Fuchs • Niels Henze • Albrecht Schmidt • Daniel Weiskopf • Steffen Koch 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungslinien		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Grundlegende Kenntnisse in Computergraphik und Bildverarbeitung (z.B. Computergraphik 051900002 und Imaging Science 051900210) 		
12. Lernziele:	Die Studierenden haben vertiefte Kenntnisse in mehreren Bereichen der Visualisierung, Computergraphik und der interaktiven Systeme erworben und können mit den erlernten Methoden wissenschaftliche Arbeiten in diesem Bereich verstehen. Sie haben das notwendige Rüstzeug, um eine Masterarbeit im Gebiet der Visualisierung, Computergraphik und der interaktiven Systeme anzufertigen.		
13. Inhalt:	Es werden Vorlesungen bzw. Vorlesungen mit Übungen zur Visualisierung und zu Interaktiven Systemen im Umfang von 8 SWS besucht, die im MINF-Katalog (MINF 1-8) den entsprechenden Verwendungshinweis tragen und dort inhaltlich beschrieben werden. In dem Vertiefungsmodul Visualisierung und Interaktive Systeme gehören hierzu die Veranstaltungen: <ul style="list-style-type: none"> • Visual Computing (Lecture and Exercise in winter semester, 4SWS) • Information Visualization (Lecture and Exercise in winter semester, 4SWS) • Multimodal Interaction for Ubiquitous Computers (Lecture and Exercise in winter semester, 4SWS) • Image Synthesis / Bildsynthese (Lecture and Exercise in summer semester, 4SWS) • Scientific Visualization (Lecture and Exercise in summer semester, 4SWS) • Geometric Modeling and Animation / Geometrische Modellierung und Animation (Lecture and Exercise in summer semester, 4SWS) 		
14. Literatur:	Andrew S. Glassner, Principles of Digital Image Synthesis, 1995		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 294001 Deutschsprachige Lehrveranstaltungen Wintersemester • 294002 Deutschsprachige Lehrveranstaltungen Sommersemester • 294003 courses in english - winter semester • 294004 courses in english - summer semester 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 84 Stunden Selbststudium: 276 Stunden		

110 Vertiefungsmodule Pflicht

Zugeordnete Module:

- 33350 Forschungsmethoden in der Softwaretechnik
- 36410 Requirements Engineering und Software-Architektur
- 42810 Software-Qualitätssicherung und -Wartung
- 42820 Prozessanalyse
- 42830 Entwicklungsprojekt
- 42860 Hauptseminar (Master SWT 1)
- 55560 Hauptseminar (Master SWT 2)

Modul: 42830 Entwicklungsprojekt

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Lars Grunske		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungsmodule Pflicht		
11. Empfohlene Voraussetzungen:	keine		
12. Lernziele:	Die Teilnehmer vertiefen ihre Erfahrungen mit den Problemen in der Entwicklung und Wartung von größeren Softwaresystemen und der Rolle der Spezifikation.		
13. Inhalt:	<ul style="list-style-type: none"> • Erstellung einer detaillierten Spezifikation • Entwicklung eines fortgeschrittenen Softwaresystems • Änderung und Erweiterung eines bestehenden Systems 		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	428301 Praktikum Entwicklungsprojekt		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 320 Stunden Nachbearbeitungszeit: 160 Stunden		
17. Prüfungsnummer/n und -name:	42831 Entwicklungsprojekt (LBP), schriftlich und mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Zuverlässige Softwaresysteme		

Modul: 33350 Forschungsmethoden in der Softwaretechnik

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Lars Grunske		
9. Dozenten:	Lars Grunske		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungsmodule Pflicht		
11. Empfohlene Voraussetzungen:	keine		
12. Lernziele:	Die Teilnehmer haben einen Überblick über die in der Softwaretechnik üblichen Forschungsmethoden und können eine Auswahl, insbesondere empirische Methoden, anwenden. Sie können statistische Methoden anwenden, um praktische Fragestellungen und Forschungsarbeiten aus der Softwaretechnik zu bearbeiten.		
13. Inhalt:	<ul style="list-style-type: none"> • Wissenschaftstheorie • Theoretische, methodische, konstruktive und empirische Forschung • Qualitative und quantitative Methoden • Systematische Literaturlauswertung, Umfragen, Interviews • Experimente und Fallstudien • Schreiben und publizieren • Deskriptive (beschreibende) Statistik • Inferentielle (induktive) Statistik • Wichtige Verteilungen und Hypothesentests • Fallbeispiele 		
14. Literatur:	<ul style="list-style-type: none"> • Shull, Singer, Sjøberg (Eds.). Guide to Advanced Empirical Software Engineering. Springer, 2008 • Leedy, Ormrod. Practical Research: Planning and Design. Pearson Prentice Hall, 2009 • Rosenkrantz. Introduction to Probability and Statistics for Scientists and Engineers. McGraw-Hill, 1997 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 333501 Vorlesung Forschungsmethoden in der Softwaretechnik • 333502 Übung Forschungsmethoden in der Softwaretechnik 		
16. Abschätzung Arbeitsaufwand:	<ul style="list-style-type: none"> • Vorlesung: Präsenzzeit: 31,5 Stunden, Nachbearbeitungszeit: 103,5 • Übung: Präsenzzeit: 10,5 Stunden, Nachbearbeitungszeit: 34,5 		
17. Prüfungsnummer/n und -name:	33351 Forschungsmethoden in der Softwaretechnik (PL), schriftlich oder mündlich, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Zuverlässige Softwaresysteme		

Modul: 42860 Hauptseminar (Master SWT 1)

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes Semester
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungsmodule Pflicht		
11. Empfohlene Voraussetzungen:	Variabel: Je nach dem gewählten Seminarthema können Vorkenntnisse aus weiteren Vorlesungen benötigt werden.		
12. Lernziele:	Die Studierenden können sich mit wissenschaftlicher Originalliteratur auseinandersetzen, deren Kernaussagen rezipieren und sich ein spezielles Thema überwiegend im Selbststudium erarbeiten. Sie sind fähig relevante Daten zu sammeln und zu interpretieren und ihre Erkenntnisse einem Fach- und Laienpublikum verständlich zu präsentieren und auf Fragen aus dem Publikum angemessen und sachgerecht zu reagieren. Sie haben gelernt, sich mit einem wissenschaftlichen Thema über einen längeren Zeitraum hinweg auseinander zu setzen und eigenständig aktuelle Hintergrundinformation zu beschaffen. Sie haben generische Kompetenzen erworben, etwa aktiv an einer wissenschaftlichen Diskussion zu einem vorher bekannten Thema teilzunehmen und durch Fragen an den Vortragenden ihr Verständnis zu erweitern. Sie können eine Diskussion leiten und moderieren und sind befähigt, ihre Ergebnisse den Seminarteilnehmern vorzustellen und mit Hilfe moderner Präsentationstechniken zu visualisieren. Sie sind in der Lage, das von ihnen erarbeitete Thema auch schriftlich darzustellen.		
13. Inhalt:	Variabel: Es werden Seminare zu diversen häufig aktuellen Themen angeboten. Welche Seminare zugelassen sind, entscheidet die Studienkommission. Zugelassene Seminare werden typischer Weise durch Aushang bekannt gegeben. Die Seminare sind in Größe und Inhalt so gestaltet, dass die generischen Kompetenzen (Schlüsselqualifikationen) der Studierenden entwickelt werden.		
14. Literatur:	Die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben.		
15. Lehrveranstaltungen und -formen:	428601 Hauptseminar		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 21 Stunden Selbststudium: 69 Stunden		
17. Prüfungsnummer/n und -name:	42861 Hauptseminar (Master SWT 1) (BSL), schriftlich und mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von:

Modul: 55560 Hauptseminar (Master SWT 2)

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:			
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungsmodule Pflicht		
11. Empfohlene Voraussetzungen:			
12. Lernziele:			
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:	555601 Hauptseminar		
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	55561 Hauptseminar (Master SWT 2) (BSL), schriftlich und mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 42820 Prozessanalyse

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Dozenten der Informatik		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungsmodule Pflicht		
11. Empfohlene Voraussetzungen:	Forschungsmethoden für die Softwaretechnik		
12. Lernziele:	Die Teilnehmer können einen Software-Entwicklungsprozess in der Praxis verstehen, analysieren, dokumentieren und Verbesserungsvorschläge angemessen präsentieren.		
13. Inhalt:	<ul style="list-style-type: none"> • Arbeit in Dreierteams • Analyse und Dokumentation eines Entwicklungsprozesses einer Firma • Interviews und Umfragen • Verbesserungsvorschläge zum Prozess • Präsentation der Ergebnisse 		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	428201 Praktikum Prozessanalyse		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 20 Stunden Nachbearbeitungszeit: 115		
17. Prüfungsnummer/n und -name:	42821 Prozessanalyse (USL), schriftlich und mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Software-Engineering		

Modul: 36410 Requirements Engineering und Software-Architektur

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Lars Grunske		
9. Dozenten:	Lars Grunske		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungsmodule Pflicht		
11. Empfohlene Voraussetzungen:	Grundlagen des Software Engineerings		
12. Lernziele:	Die Teilnehmer haben einen umfassenden Überblick über die verfügbaren Methoden und Techniken zum Requirements Engineering und zur Software-Architektur. Sie haben vertiefte Anwendungserfahrung in ausgewählten Methoden und Techniken.		
13. Inhalt:	<ul style="list-style-type: none"> • Methoden des Requirements Engineerings • Beschreibung und Modellierung von Anforderungen • Analyse und Validierung von Anforderungen • Management von Anforderungen • Modellierung, Erstellung und Analyse von Software-Architekturen • Architekturmuster • Requirements Engineering und Architektur im Entwicklungsprozess 		
14. Literatur:	<ul style="list-style-type: none"> • Robertson, Robertson. Mastering the Requirements Process. Addison-Wesley Professional, 2006 • Sommerville, Sawyer. Requirements Engineering: A Good Practice Guide. John Wiley & Sons, 1997 • Bass, Clements, Kazman. Software Architecture in Practice, 2nd edition. Addison-Wesley Professional, 2003 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 364101 Vorlesung Requirements Engineering und Software-Architektur • 364102 Übung Requirements Engineering und Software-Architektur 		
16. Abschätzung Arbeitsaufwand:	<ul style="list-style-type: none"> • Vorlesung Requirements Engineering und Software-Architektur: Präsenzzeit: 31,5 Stunden, Nachbearbeitungszeit: 103,5 • Übung Requirements Engineering und Software-Architektur: Präsenzzeit: 10,5 Stunden, Nachbearbeitungszeit: 34,5 		
17. Prüfungsnummer/n und -name:	36411 Requirements Engineering und Software-Architektur (PL), schriftlich oder mündlich, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Zuverlässige Softwaresysteme		

Modul: 42810 Software-Qualitätssicherung und -Wartung

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Stefan Wagner		
9. Dozenten:	Stefan Wagner		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Vertiefungsmodule → Vertiefungsmodule Pflicht		
11. Empfohlene Voraussetzungen:	Grundlagen des Software Engineerings oder Software Engineering für Softwaretechniker		
12. Lernziele:	Die Teilnehmer verstehen Qualitätssicherungs- und Wartungsprozesse für Software. Sie haben einen umfassenden Überblick über entsprechende Methoden und Techniken, die sie auch einordnen können. Für eine Auswahl davon haben sie vertiefte Anwendungskennnisse.		
13. Inhalt:	<ul style="list-style-type: none"> • Qualitätssicherungs- und Wartungsprozesse • Prozesskontrolle und -steuerung • Qualitätssicherungsmethoden und Qualitätsmodelle • Vorhersagemodelle, Repository Mining, Software Analytics • Programmanalyse und Programmverstehen • Werkzeugunterstützung 		
14. Literatur:	<ul style="list-style-type: none"> • Wagner. Software Product Quality Control. Springer, 2013 • Liggesmeyer. Software-Qualität. Testen, Analysieren und Verifizieren von Software. Spektrum Akademischer Verlag, 2002 • Sneed, Hasitschka, Teichmann. Software-Produktmanagement. Wartung und Weiterentwicklung bestehender Anwendungssysteme. Dpunkt, 2004 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 428101 Vorlesung Software-Qualitätssicherung und -Wartung • 428102 Übung Software-Qualitätssicherung und -Wartung 		
16. Abschätzung Arbeitsaufwand:	<ul style="list-style-type: none"> • Vorlesung: Präsenzzeit: 31,5 Stunden, Nachbearbeitungszeit: 103,5 • Übung: Präsenzzeit: 10,5 Stunden, Nachbearbeitungszeit: 34,5 		
17. Prüfungsnummer/n und -name:	42811 Software-Qualitätssicherung und -Wartung (PL), schriftlich oder mündlich, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Software-Engineering		

200 Spezialisierungsmodule - MSWT

Zugeordnete Module:	10040	Bildsynthese
	10250	Parallele Systeme
	11900	Design and Test of Systems-on-a-Chip
	24900	Fachpraktikum Rechnergestützte Methoden der Produktentwicklung
	29430	Computer Vision
	29440	Geometric Modeling and Computer Animation
	29450	Graphentheorie
	29460	Kryptographische Verfahren
	29470	Machine Learning
	29480	Loose Coupling and Message Based Applications
	29500	Visual Computing
	29510	Service Computing
	29550	Algorithmische Geometrie
	29560	Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems
	29570	Computer Interface Technologien
	29580	Data Compression
	29590	Digitale Systeme
	29600	Digital System Design II
	29610	Hardware Based Fault Tolerance
	29630	Konzepte der Programmiersprachen
	29640	Mikrocontroller
	29650	Parallele Programmierung
	29660	Programmanalysen und Compilerbau
	29670	Rapid Prototyping
	29680	Real-Time Programming
	29690	Real-Time Video Processing I
	29700	Real-Time Video Processing II
	29710	Embedded Systems Engineering
	29720	Mobile Computing
	29730	Modelling, Simulation, and Specification
	29740	Fachpraktikum Eingebettete Systeme
	39250	Distributed Systems I
	40680	Optimization
	42420	High Performance Computing
	42480	Ausgewählte Kapitel des Wissenschaftlichen Rechnens
	42520	Services and Service Composition
	42900	Business Process Management
	42910	Advanced Business Process Management
	42920	Hardware-Software-Codesign
	45730	Distributed Systems II
	45740	Rechnernetze II
	45750	Fachpraktikum Verteilte Systeme
	46660	Service Management and Cloud Computing, and Evaluation
	46760	Theoretical and Methodological Foundations of Visual Computing
	48500	Image Synthesis
	48570	Practical Course Visual Computing
	48580	Reinforcement Learning
	48600	Robotics I
	48620	Scientific Visualization
	48640	Theoretical and Methodological Foundations of Autonomous Systems
	51720	IT-Strategy
	51740	Quantencomputing
	55600	Advanced Information Management

55620 Data Warehousing, Data Mining, and OLAP
55630 Information Visualization and Visual Analytics
55640 Correspondence Problems in Computer Vision
55650 Multimodal Interaction for Ubiquitous Computers
55740 Advanced Service Computing
56550 Software Verification
56680 Automaten über unendlichen Objekten
56790 Parallele Numerik
56980 Fachpraktikum Cloud Architekturen und Management
57050 Compilerbau
58190 Entwurf und Implementierung eines Compilers

Modul: 42910 Advanced Business Process Management

2. Modulkürzel:	052010007	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> • Frank Leymann • Dimka Karastoyanova 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	052010006 Workflow Management 1		
12. Lernziele:	Am Ende der Veranstaltungen haben die Teilnehmer weiterführende Ansätze zur Modellierung von Prozessen und zur Spezifikation von Workflows verstanden. Die Rolle von Muster in der Beschreibung von Workflows ist klar geworden. Verfahren des Process Mining sind theoretisch dargestellt. Die Notwendigkeit zur P2P-Verzahnung ("Choreographien") von Prozessen und entsprechende Ansätze sind klar. Ebenso verstanden ist das darüber hinausgehende Konzept der Komponentenverdrahtung. Weitere Architekturen und Einsatzgebiete von WFMS sind verstanden.		
13. Inhalt:	In der Vorlesung werden fortgeschrittene Themen des Workflowmanagement vorgestellt. Aktuelle Entwicklungen aus dem Forschungsumfeld und der Industrie auf dem Gebiet werden diskutiert. Human Task Management Weitere Ansätze zur Prozessmodellierung (Pi-Kalkül, WSFL, XLANG,...) Muster (Kontrol-, Datenfluss, Organisatorisch) Process Monitoring Process Mining Peer-to-Peer Verzahnung von Prozessen (Choreographie, Gebrauchsanweisungen,...) Verdrahten von Komponenten (Global Models,...) Anwendungsbereiche (Manufakturing, Compliance,...) Prozessadaption und -flexibilität		
14. Literatur:	W. van der Aalst, K. van Hee, Workflow Management, 2002		
15. Lehrveranstaltungen und -formen:	429101 Vorlesung mit Übungen, Workflow Management 2		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 42911 Advanced Business Process Management (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Architektur von Anwendungssystemen		

Modul: 55600 Advanced Information Management

2. Modulkürzel:	051200099	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	PD Holger Schwarz		
9. Dozenten:	<ul style="list-style-type: none"> • Holger Schwarz • Bernhard Mitschang 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Lecture "Modellierung" or comparable course		
12. Lernziele:	The students learn current concepts for modeling, developing and processing database-oriented applications. Extensions to relational systems as well as non-relational systems are considered. Processing XML data is important for many application areas today. Hence, technologies and standards for XML processing and their integration into database systems constitute another focus of this course.		
13. Inhalt:	Among the topics to be discussed in this course are: - XML and database technology (XML modeling, XML storage, XML query languages, XML processing) - Content management (Enterprise content management, information retrieval, search technologies) - NoSQL data management (Key value stores, triple stores, MapReduce)		
14. Literatur:	<ul style="list-style-type: none"> • A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 2002 • H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems. The Complete Book, 2003 Will be announced at the beginning of the lecture		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 556001 Vorlesung Advanced Information Management • 556002 Übung Advanced Information Management 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	55601 Advanced Information Management (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Datenbanken und Informationssysteme		

Modul: 55740 Advanced Service Computing

2. Modulkürzel:	052010005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	5.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Dimka Karastoyanova		
9. Dozenten:	<ul style="list-style-type: none"> • Dimka Karastoyanova • Frank Leymann 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	<p>Service Computing, Lecture and Exercise (4 SWS) or Services and Service Composition, Lecture and Exercise (4SWS)</p>		
12. Lernziele:	<p>This module comprises two lectures and therefore topics from two areas of advanced service computing. The focus of the Lecture Advanced Service Computing is concepts and technologies for describing and providing stateful resources as Web Services as well as the use of Semantics in Web Services and service compositions. The focus in the Lecture Services and Security is on security aspects of service-based applications.</p>		
13. Inhalt:	<p>This module comprises two lectures and therefore topics from two areas of advanced service computing.</p> <p>Based on the topics discussed in the lecture Service Computing, in the Lecture Advanced Service Computing we will focus on concepts and technologies for describing and providing stateful resources as Web Services. In this respect we will also consider Grid Services and infrastructures. In addition, the topics Semantic Web, Ontologies and Semantic Web Services will be presented in detail. Particular attention will be paid to Semantic Web Service Technologies and frameworks like OWL-S, WSMO, SAWSDL and approaches for their use in service compositions.</p> <p>The focus in the Lecture Services and Security is on security aspects of service-based applications. Foundations of Security in enterprise architectures will be presented, as well as best practices for enterprise and IT security in terms of patterns. Basic Security approaches (e.g. prevention, detection, reaction) and mechanisms (access control, authentication, identification, cryptography) will be presented in detail. We will also discuss current state of the art of Web application and Web Service security.</p>		
14. Literatur:	<ul style="list-style-type: none"> • Literatur, die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben. • S. Graham, D. Davis, S. Simeonov, G. Daniels, P. Brittenham, Y. Nakamura, P. Fremantle, D. König, C., Building Web Services with Java (2nd Edition), 2005 • S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, Web Services Platform Architecture, 2005 		

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

15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 557401 Advanced Service Computing Lecture (Summer)• 557402 Lecture Services and Security (Winter)
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 60 Stunden Selbststudium: 120 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 55741 Advanced Service Computing (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0, Mündliche Prüfung von 30 Min• V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Architektur von Anwendungssystemen

Modul: 29550 Algorithmische Geometrie

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

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

2. Modulkürzel:	051700024	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	10310 Rechnerorganisation oder 10140 Grundlagen der Rechnerarchitektur / Advanced Processor Architecture		
12. Lernziele:	Knowledge of the most important algorithms and methods in design automation tools at any design level		
13. Inhalt:	Firstly, the lecture points out the basic algorithms in modern design automation software. Next, the problems occurring in synthesis, analysis and test of digital circuits at the different design levels are discussed and their solutions are mapped to the basic algorithms. Major aspects in the discussion are the challenges and problems arising from nanometer technology. Here the focus always lies on the software supporting the design of digital systems.		
14. Literatur:	<ul style="list-style-type: none"> • G. De Micheli: Synthesis and Optimization of Digital Circuits, McGrawHill, New York, NY, USA, 1994. • Hachtel, G. D. and Somenzi: Logic Synthesis and Verification Algorithms, Kluwer Academic Publishers, Norwell, MA, USA, 2000. • Ban Wong, Anurag Mittal, Yu Cao: Nano-CMOS Circuit and Physical Design, John Wiley & Sons Inc, 2004. • Ashish Srivastava, Dennis Sylvester, David Blaauw: Statistical Analysis and Optimization for VLSI: Timing and Power, Springer, 2005. • Jens Lienig: Layoutsynthese elektronischer Schaltungen - Grundlegende Algorithmen fuer die Entwurfsautomatisierung, Springer, 2006. • L.-T. Wang, Y.-W. Chang, K.-W. Cheng: Electronic Design Automation, Morgan Kaufmann, 2009 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 295601 Vorlesung Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems • 295602 Übung Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	
17. Prüfungsnummer/n und -name:	29561 Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems (PL), schriftlich oder mündlich,		

90 Min., Gewichtung: 1.0, Klausur 90 min. oder mündliche
Prüfung 30 min.

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Technische Informatik

Modul: 42480 Ausgewählte Kapitel des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240030	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> • Dirk Pflüger • Stefan Zimmer • Miriam Mehl 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Katalog ISW <p>B.Sc. Softwaretechnik, PO 2012</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Katalog ISW <p>B.Sc. Softwaretechnik, PO 2012</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Wahlmodule aus Master SWT <p>B.Sc. Softwaretechnik, PO 2012</p> <ul style="list-style-type: none"> → Vorgezogene Master-Module <p>M.Sc. Softwaretechnik, PO 2012</p> <ul style="list-style-type: none"> → Spezialisierungsmodule - MSWT 		
11. Empfohlene Voraussetzungen:	<p>080300100 Mathematik für Informatiker und Softwaretechniker und 051240005 Numerische und Stochastische Grundlagen der Informatik bzw. 051240006 Einführung in die Numerik und Stochastik für Softwaretechniker 051240020 Grundlagen des wissenschaftlichen Rechnens</p>		
12. Lernziele:	Die Teilnehmer kennen ausgewählte aktuelle Forschungsthemen des wissenschaftlichen Rechnens und können mit der zugehörigen Primärliteratur arbeiten.		
13. Inhalt:	Aktuelle weiterführende Forschungsthemen des wissenschaftlichen Rechnens, wie z.B. adaptive Finite Elemente, hierarchische Basen und dünne Gitter, robuste Multilevellöser, Wavelets und schnelle Wavelettransformation, p-Version oder Spektralverfahren.		
14. Literatur:	<p>Primärliteratur zu den behandelten Themen:</p> <ul style="list-style-type: none"> • Bungartz/Griebel: Sparse Grids; Acta Numerica, Volume 13, p. 147-269 • Quarteroni/Valli: Numerical approximation of partial differential equations • Quarteroni: Numerical models for differential problems 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 424801 Vorlesung Ausgewählte Kapitel des Wissenschaftlichen Rechnens • 424802 Übung Ausgewählte Kapitel des Wissenschaftlichen Rechnens 		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden Selbststudiumszeit: 138 Stunden</p>		

17. Prüfungsnummer/n und -name: 42481 Ausgewählte Kapitel des Wissenschaftlichen Rechnens (PL),
schriftlich oder mündlich, 90 Min., Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Simulation großer Systeme

Modul: 56680 Automaten über unendlichen Objekten

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

Logik erster Stufe und andere Fragmente von MSO

Paritätsspiele

Automaten über unendlichen Bäumen

Rabins Baumtheorem

14. Literatur:	<ul style="list-style-type: none"> • Volker Diekert, Manfred Kufleitner, Gerhard Rosenberger: Diskrete algebraische Methoden: Arithmetik, Kryptographie, Automaten und Gruppen. De Gruyter, Berlin 2013. • Volker Diekert und Paul Gastin: First-order definable languages. In Jörg Flum, Erich Grädel, Thomas Wilke (eds.). Logic and Automata: History and Perspectives. Texts in Logic and Games 2, Amsterdam University Press 2008, pp. 261-306. • Wolfgang Thomas: Automata on infinite objects. In Jan van Leeuwen (ed.). Handbook of Theoretical Computer Science, volume B: Formal Models and Semantics. Elsevier, 1990, pp. 133-192. • Wolfgang Thomas: Languages, Automata, and Logic. In Grzegorz Rozenberg and Arto Salomaa (eds). Handbook of Formal Languages, volume 3: Beyond Words. Springer, New York, 1997, pp. 389-455.
15. Lehrveranstaltungen und -formen:	566801 Vorlesung Automaten über unendlichen Objekten
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden
17. Prüfungsnummer/n und -name:	56681 Automaten über unendlichen Objekten (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

Modul: 10040 Bildsynthese

2. Modulkürzel:	051900012	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> • Martin Fuchs • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009, 6. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012, 6. Semester → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051900002 Computergraphik 		
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. Darüber hinaus kennen sie interaktive Verfahren, die unter Ausnutzung programmierbarer Grafik-Hardware realistische Beleuchtungseffekte in Echtzeit approximieren können, sowie bildbasierte Ansätze, die ohne geometrische Daten realistische Darstellungen erzeugen. Bild-basierte Verfahren verzichten auf eine geometrische Repräsentation der Szene und erzeugen neue Ansichten aus anderen aufgenommenen Bildern.</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 • Bild-basiertes Rendering 		
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:	<ul style="list-style-type: none"> • 10041 Bildsynthese (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich, Übungsschein.
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme

Modul: 42900 Business Process Management

2. Modulkürzel:	052010006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> • Frank Leymann • Dimka Karastoyanova 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	611 Grundlagen der Architektur von Anwendungssystemen, Vorlesung mit Übung, 4,0 SWS		
12. Lernziele:	<p>The course has the objective to provide knowledge about the essential modelling constructs for workflows and their mapping to corresponding workflow languages. In addition, the life cycle of Workflow-based applications will be presented in detail and connected to the Architecture of Workflow Management Systems, which will also be presented. Moreover, the goal is to enable students to use workflow languages (in particular BPEL) in practice. In this respects students will also understand the fundamental approach process graphs, which is applied in workflow languages. Of great importance are , mechanisms for fault handling and exception handling - these will be explained in detail and students will be able to apply them.</p>		
13. Inhalt:	<p>Workflows are IT realisations of business processes and are also considered an approach of significant importance for composition of applications. This course will introduce the foundations of this area, also known as Business Process Management BPM).</p> <ol style="list-style-type: none"> 1. Historical Development of the Workflow Technology 2. Business Re-engineering (BPM Lifecycle, Tools,...) 3. Architecture of WFMS (Navigator, Executor, Worklist Manager,...) 4. Flow Languages (FDL, BPEL) 5. Process Model Graph (mathematical meta-model: syntax, operational semantics) 6. Advanced functions (sub-processes, event handling, instance modifications, adaptation) 7. Two-level programming paradigm 8. Transactional support in workflows 		
14. Literatur:	<ul style="list-style-type: none"> • F. Leymann, D. Roller, Production Workflow, 2000 • W. van der Aalst, K. van Hee, Workflow Management, 2002 		
15. Lehrveranstaltungen und -formen:	429001 Vorlesung mit Übungen, Workflow Management 1		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 42901 Business Process Management (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0 		

-
- V Vorleistung (USL-V), schriftlich, eventuell mündlich
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Architektur von Anwendungssystemen

Modul: 57050 Compilerbau

2. Modulkürzel:	051510TBD	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:			
12. Lernziele:			
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 570501 Vorlesung Compilerbau • 570502 Übung Compilerbau 		
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	57051 Compilerbau (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 29570 Computer Interface Technologien

2. Modulkürzel:	051230105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Kenntnisse in mindestens einem Fach der Technischen Informatik oder ähnlichen Fächern und Erfahrungen in mindestens einer Programmiersprache.		
12. Lernziele:	Die Studierenden haben die Funktionsweise und den Aufbau von Coputer Interfaces verstanden. Sie beherrschen verschiedene Interface-Konzepte und kennen die Eigenschaften der Datenströme wie Latenzzeit, tatsächliche Durchsatzrate, Echtzeitfähigkeit, Umgang mit Übertragungsfehlern, etc.		
13. Inhalt:	<ul style="list-style-type: none"> • Grundlagen - Computer Interfaces • Computer Interfaces und OSI-Modelle • Bus- und Netz-Topologien • Line und Error Codes • Protokolle • Treiber • Compliance Tests • Standardization Groups: USB, PCI, etc. 		
14. Literatur:	<ul style="list-style-type: none"> • Patterson, David A.; Hennessey, John L., Computer Organization and Design - The Hardware / Software Interface, 2008 <p>More literature is named in the lecture.</p>		
15. Lehrveranstaltungen und -formen:	295701 Vorlesung mit Übung Computer Interface Technologien		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	29571 Computer Interface Technologien (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Minuten oder mündliche Prüfung von 30 Min.		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 29430 Computer Vision

2. Modulkürzel:	051900215	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Andrés Bruhn		
9. Dozenten:	Andrés Bruhn		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 080300100 Mathematik für Informatiker und Softwaretechniker • Modul 050700005 Imaging Science 		
12. Lernziele:	<p>Der Student / die Studentin beherrscht die Grundlagen der Merkmalsextraktion und -repräsentation, des 3-D Maschinensehens, der Bildsegmentierung sowie der Mustererkennung. Er/sie kann Probleme aus dem Fachgebiet einordnen und diese selbständig mit den erlernten Algorithmen und Verfahren lösen.</p> <p>The student knows the basics of feature extraction and representation, 3-D computer vision, image segmentation and pattern recognition. He/she can solve problems of the field using the methods discussed in the course.</p>		
13. Inhalt:	<ul style="list-style-type: none"> • Lineare Diffusion, Skalenräume • Bildpyramiden, Kanten und Eckendetektion • Hough-Transformation, Invarianten • Texturanalyse • Scale Invariant Feature Transform (SIFT) • Bildfolgenanalyse: lokale Verfahren • Bewegungsmodelle, Objektverfolgung, Feature Matching • Bildfolgenanalyse: globale Verfahren • Kamerageometrie, Epipolargeometrie • Stereo Matching und 3-D Rekonstruktion • Shape-from-Shading • Isotrope und anisotrope nichtlineare Diffusion • Segmentierung mit globalen Verfahren • Kontinuierliche Morphologie, Schockfilter • Mean Curvature Motion • Self-Snakes, Aktive Konturen • Bayes'sche Entscheidungstheorie der Mustererkennung • Klassifikation mit parametrischen Verfahren, Dichteschätzung • Klassifikation mit nicht-parametrischen Verfahren • Dimensionsreduktion <ul style="list-style-type: none"> • Linear Diffusion, Scale Space • Image Pyramids, Edges and Corners • Hough Transform, Invariants • Texture Analysis • Scale Invariant Feature Transform 		

	<ul style="list-style-type: none"> • Image Sequence Analysis: Local Methods • Motion Models, Tracking, Feature Matching • Image Sequence Analysis: Variational Methods • Camera Geometry, Epipolar Geometry • Stereo Matching and 3-D Reconstruction • Shape-from-Shading • Isotropic and Anisotropic Nonlinear Diffusion • Segmentation with Global Methods • Continuous Scaled Morphology, Shock Filters • Mean Curvature Motion • Self-Snakes, Active Contours • Bayes Decision Theory for Pattern Recognition • Classification with Parametric Techniques, Density Estimation • Classification with Non-Parametric Techniques • Dimensionality Reduction
14. Literatur:	<ul style="list-style-type: none"> • Forsyth, David and Ponce, Jean, Computer Vision. A Modern Approach.: A Modern Approach Computer Vision. A Modern Approach, 2003 • Bigun, J.: Vision with Direction, 2006 • L. G. Shapiro, G. C. Stockman, Computer Vision, 2001 • O. Faugeras, Q.-T. Luong: The Geometry of Multiple Images, 2001.
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 294301 Vorlesung Computer Vision • 294302 Übung Computer Vision
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 29431 Computer Vision (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben • V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	55640 Correspondence Problems in Computer Vision
19. Medienform:	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme

Modul: 55640 Correspondence Problems in Computer Vision

2. Modulkürzel:	051900211	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	6.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Andrés Bruhn		
9. Dozenten:	Andrés Bruhn		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 080300100 Mathematik für Informatiker und Softwaretechniker • Modul 050700005 Imaging Science • Modul 051900215 Computer Vision 		
12. Lernziele:	<p>Der Student kann Korrespondenzprobleme im Computer-Vision-Bereich selbständig einordnen, Lösungsstrategien mathematisch modellieren und diese dann geeignet algorithmisch umsetzen.</p> <p>The student has knowledge on the different correspondence problems in computer vision, is able to develop mathematical models for solution strategies and implement the corresponding algorithms in an appropriate way.</p>		
13. Inhalt:	<ul style="list-style-type: none"> • Basisverfahren: Block Matching, Detektion von Verdeckungen, Merkmalsfindung, Feature Matching • Optischer Fluss: Lokale und Globale differentiale Verfahren, Parametrisierungsmodelle, Konstanzannahmen, Daten- und Glattheitsterme, Numerik, Große Verschiebungen, Hochgenaue Verfahren • Stereorekonstruktion: Projektive Geometrie, Epipolargeometrie, Schätzung der Fundamentalmatrix • Szenenfluss: Gemeinsame Schätzung von Struktur, Bewegung und Geometrie • Medizinische Bildregistrierung: Mutual Information, Elastische und krümmungsbasierte Regularisierung, Landmarks • Particle Image Velocimetry: Div-Curl-Regularisierung, Inkompressibler Navier Stokes Prior • Basic Approaches: Block Matching, Occlusion Detection, Interest Points, Feature Matching • Optic Flow: Local and Global Differential Methods, Parametrisation Models, Constancy Assumptions, Data and Smoothness Terms, Numerics, Large Displacements, High Accuracy Methods • Stereop Matching: Projective Geometry, Epipolar Geometry, Estimation of the Fundamental Matrix • Scene Flow: Joint Estimation of Structure, Motion, and Geometry • Medical Image Registration: Mutual Informaion, Elastic and Curvature-Based Regularisation, Landmarks 		

	<ul style="list-style-type: none"> • Particle Image Velocimetry: Div-Curl-Regularisation, Incompressible Navier Stokes Prior
14. Literatur:	<ul style="list-style-type: none"> • O. Faugeras, Q.-T. Luong: The Geometry of Multiple Images, 2001. • J. Modersitzki: Numerical Methods for Image Registration, 2003. • A. Bruhn: Variational Optic Flow Computation: Accurate Modeling and Efficient Numerics, Ph.D. Thesis, 2006.
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 556401 Vorlesung Correspondence Problems in Computer Vision • 556402 Übung Correspondence Problems in Computer Vision
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 55641 Correspondence Problems in Computer Vision (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben • V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme

Modul: 29580 Data Compression

2. Modulkürzel:	051230110	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	This course requires basic knowledge in mathematics.		
12. Lernziele:	The students learn the concepts of data compression and acquire an understanding of different algorithms for data compression. Furthermore they will be able to implement and further develop the algorithms discussed in the course.		
13. Inhalt:	<ul style="list-style-type: none"> • Shannon Entropy • Huffman coding • Universal codes • Arithmetic coding • Lossy and Lossless compression • Image data compression • Dictionary based compression 		
14. Literatur:	<ul style="list-style-type: none"> • Khalid Sayood, Introduction to Data Compression, 2005 • More literature is named in the lecture 		
15. Lehrveranstaltungen und -formen:	295801 Vorlesung mit Übung Datenkompression		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	29581 Data Compression (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0, written 90 Min. or oral 30 Min.		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 55620 Data Warehousing, Data Mining, and OLAP

2. Modulkürzel:	051210105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Bernhard Mitschang		
9. Dozenten:	<ul style="list-style-type: none"> • Bernhard Mitschang • Holger Schwarz 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Lecture "Modellierung" or comparable course		
12. Lernziele:	After attending this lecture, students understand the challenges behind the integration of heterogeneous data sources in consolidated warehouses and the provisioning of analytical services. They know the typical data warehouse architecture as well as current trends, e.g., real-time data warehousing. Further topics are the structure of a data warehouse and the main processes for building data warehouses (extraction, transformation, load). A special focus is on technologies to analyze data warehouse data, e.g., reporting, online analytic processing and data mining, and their role as part of analytical services.		
13. Inhalt:	Among the topics to be discussed in this course are: <ul style="list-style-type: none"> - Introduction to data warehousing - Data warehouse architecture - Data warehouse design - Extraction, transformation, load - ETL as a service - Introduction to analytics and analytic services - Real-time reporting - Online analytic processing - Data mining 		
14. Literatur:	<ul style="list-style-type: none"> • A. Kemper, A. Eickler, Datenbanksysteme - Eine Einführung, 2004 • H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems. The Complete Book, 2003 Will be announced at the beginning of the lecture		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 556201 Vorlesung Data Warehousing, Data Mining und OLAP-Technologien • 556202 Übung Data Warehousing, Data Mining und OLAP-Technologien 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 55621 Data Warehousing, Data Mining, and OLAP (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Übungsleistungen während der Unterrichtsperiode als Prüfungsvoraussetzung. 		

-
- V Vorleistung (USL-V), schriftlich, eventuell mündlich, 60 Min.

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Datenbanken und Informationssysteme

Modul: 11900 Design and Test of Systems-on-a-Chip

2. Modulkürzel:	051700015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	Hans-Joachim Wunderlich		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012, 5. Semester → Ergänzungsmodule → Wahlmodule aus Master SWT B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012, 5. Semester → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051700005 Rechnerorganisation • Modul 051700010 Grundlagen der Rechnerarchitektur 		
12. Lernziele:	<p>The students of this course have gained a basic understanding of development and test of complex embedded hardware / software systems. The participants have become acquainted with the essential steps of synthesis, validation, test and programming and have learned, how to use the related tools for design automation.</p> <p>Besides the different design styles, paradigms and standards, the essential steps of automated design, test and programming of digital and mixed signal circuits have been discussed. Exercises and labs have led to practical insight into the design flow and commercial design automation tools.</p>		
13. Inhalt:	<p>The course comprises:</p> <ul style="list-style-type: none"> • Overview of system design • IP core reuse • Standards and platforms • Elements of analog and mixed signal design • Design validation and verification • Test and design for testability with the related standards • Application and programming of embedded processors 		
14. Literatur:	<ul style="list-style-type: none"> • A. Sloss, D. Symes, C. Wright: ARM System Developer's Guide: Designing and Optimizing System Software, 2004 • L.-T. Wang, C.-W. Wu, X. Wen: VLSI Test Principles and Architectures - Design for Testability, 2006 • M. Keating, P. Bricaud: Reuse Methodology Manual for System-on-a-Chip Designs, 2007 • M. L. Bushnell, V. D. Agrawal: Essentials of Electronic Testing, 2005 • S. Furber: ARM System-on-Chip Architecture, 2000 • W. Wolf: Modern VLSI Design: System-on-Chip Design, 2002 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 119001 Vorlesung Design and Test of Systems on a Chip 		

- 119002 Übung Design and Test of Systems on a Chip
- 119003 Praktikum Design and Test of Systems on a Chip

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden
Nachbearbeitungszeit: 138 Stunden

Gesamt: 180 Stunden

17. Prüfungsnummer/n und -name:

- 11901 Design and Test of Systems-on-a-Chip (LBP), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0
- V Vorleistung (USL-V), schriftlich, eventuell mündlich, 90 Min.

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Institut für Technische Informatik

Modul: 29600 Digital System Design II

2. Modulkürzel:	051230122	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	This lectures requires the knowledge of "System Design I". Alternatively, knowledge of "Technische Informatik" is sufficient to follow the course.		
12. Lernziele:	The students will learn to build and implement a complex digital system by using digitals components on a circuit board, and will acquire an in-depth knowledge for implementing complex digital systems using FPGA's.		
13. Inhalt:	<ul style="list-style-type: none"> • Presentation of a case study of a digital system • Simulatable specification of the system • Architecture for Implementation using FPGAs • Design and design tools for board integration • Implementation of a digital system • Verification of a digital system 		
14. Literatur:	<ul style="list-style-type: none"> • Kou-Chuan Chang, K. C. Chang, Digital Systems Design with VHDL and Synthesis: An Integrated Approach, 1999 • More literature is named in the lecture 		
15. Lehrveranstaltungen und -formen:	296001 Vorlesung mit Übung Digital System Design II		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29601 Digitale Systeme II (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 29590 Digitale Systeme

2. Modulkürzel:	051230120	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Kenntnisse in einem Fach aus der Technischen Informatik oder einem ähnlichen Gebiet.		
12. Lernziele:	Die Studierende beherrschen den Entwurf Digitaler Systeme durch die Integration von digitalen Komponenten auf einem Board und die Realisierung von digitaler Komponenten mittels FPGAs.		
13. Inhalt:	<ul style="list-style-type: none"> • Praktische Einführung in den System-Entwurf mit digitalen Komponenten wie Schnittstellenbausteinen zur Kommunikation, FPGAs, Prozessoren, intelligenten Sensoren etc. • Einführung und Verwendung der Hardware-Beschreibungssprache VHDL zum Entwurf Digitaler Systeme • Digitale Systeme und Board-Integration von digitalen Komponenten • Aufbau von Computer-Boards u. Gbit/s-Interconnects • Entwurf auf höheren Abstraktionsebenen zur schnellen Entwicklung von Prototypen 		
14. Literatur:	<ul style="list-style-type: none"> • Kou-Chuan Chang, K.C. Chang, Digital Systems Design with VHDL and Synthesis: An Integrated Approach, 1999 <p>More literature is named in the lecture.</p>		
15. Lehrveranstaltungen und -formen:	295901 Vorlesung mit Übung Digital System Design I		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 29591 Digitale Systeme (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündliche Prüfung von 30 Min. • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 120 Min. 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 39250 Distributed Systems I

2. Modulkürzel:	051200015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	Kurt Rothermel		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2009, 5. Semester → Ergänzungsmodule → Katalog ISG B.Sc. Softwaretechnik, PO 2012, 5. Semester → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012, 5. Semester → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Programmierung und Software-Entwicklung Datenstrukturen und Algorithmen Systemkonzepte und -Programmierung		
12. Lernziele:	The Students will gain an understanding of the basic characteristics, concepts and methods of distributed systems. Furthermore, the ability to analyze existing distributed applications and platforms with regard to its specific properties will be obtained. The implementation of distributed applications as well as system platforms based on the shown methods of that course is another objective. Due to the knowledge provided in that course, the students will be able to communicate with other experts of other professional disciplines, about topics in the field of distributed systems.		
13. Inhalt:	<ol style="list-style-type: none"> 1. Introduction to distributed systems 2. System models 3. Communication: Messages, Remote Procedure Call (RPC), Remote Method Invocation RMI 4. Naming: Generating and Resolution 5. Time Management and clocks in distributed Systems: Applications, logical clocks, physical clocks, synchronization of clocks 6. Global state: concepts, snapshot algorithms, distributed Debugging 7. Transaction management: Serializability, barrier methods, 2-phase-commit-protocols 8. Data replication: primary copy, consensus-protocols and other algorithms 9. Safety/Security: Methods for confidentiality, integrity, authentication and authorization 10. Multicast-algorithms: processing model, broadcast-semantics and algorithms 		
14. Literatur:	Literatur, siehe Webseite zur Veranstaltung		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 392501 Vorlesung Verteilte Systeme • 392502 Übungen Verteilte Systeme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 h Selbststudiumszeit / Nachbearbeitungszeit: 138 h Gesamt: 180 h		

Modul: 45730 Distributed Systems II

2. Modulkürzel:	051200169	5. Moduldauer:	2 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:		Univ.-Prof. Kurt Rothermel	
9. Dozenten:		Kurt Rothermel	
10. Zuordnung zum Curriculum in diesem Studiengang:		B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT	
11. Empfohlene Voraussetzungen:		The Lecture requires basic knowledge from the course Distributed Systems I	
12. Lernziele:		In this lecture, the acquired knowledge from the previous lecture "Verteilte Systeme I" is deepened. The student will gain information about further practice-oriented problems and will implement protocols to solve those problems. The student will be capable to analyze distributed systems in terms of these problems, design, apply and develop protocols for specific applications.	
13. Inhalt:		<ol style="list-style-type: none"> 1. Group communication 2. Consensus 3. Fault tolerant services 4. Wave algorithms 5. Termination 6. Garbage collection 7. Election 8. Deadlocks 9. Organisational & Introduction 	
14. Literatur:		<ul style="list-style-type: none"> • J.L. Welch, H. Attiya, Distributed Computing: Fundamentals, Simulations and Advanced Topics, 1997 The event is based on a collection of scientific papers, which will be announced in the lecture.	
15. Lehrveranstaltungen und -formen:		<ul style="list-style-type: none"> • 457301 Vorlesung Verteilte Algorithmen • 457302 Vorlesung Asynchronous Middleware Systems 	
16. Abschätzung Arbeitsaufwand:		Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden	
17. Prüfungsnummer/n und -name:		45731 Distributed Systems II (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0	
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:		Verteilte Systeme	

Modul: 29710 Embedded Systems Engineering

2. Modulkürzel:	051711027	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	keine		
12. Lernziele:	Master-level understanding of the design methodology and advanced design techniques for constructing and analyzing embedded hardware / software systems.		
13. Inhalt:	<ol style="list-style-type: none"> 1. Introduction to embedded systems and their design constraints 2. Synthesis models and algorithms 3. System level synthesis 4. High level synthesis 5. Pipelined data path and controller design 6. Software task scheduling and schedulability analysis 7. Static and dynamic methods for scheduling and priority assignment 8. Communication architectures for embedded systems 		
14. Literatur:	Skript „Embedded Systems Engineering“ G. Buttazzo: Hard Real Time Computing Systems. 2nd edition, Springer, 2005 P. Eles, K. Kuchcinski, Z. Peng: System Synthesis with VHDL. Kluwer Academic Publishers, 1998. P. Marwedel: Embedded Systems Design. Springer, 2006		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 297101 Vorlesung Embedded Systems Engineering • 297102 Übung Embedded Systems Engineering 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Summe: 180 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 29711 Embedded Systems Engineering (Klausur) (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich, Als Zulassungsvoraussetzung zur Klausur ist folgende Vorleistung zu erbringen: Teilnahme an den Übungen, Präsentation der Lösung wenigstens einer Aufgabe. 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)		

Modul: 58190 Entwurf und Implementierung eines Compilers

2. Modulkürzel:	05151313	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	<ul style="list-style-type: none"> • Erhard Plödereder • Timm Felden 		
10. Zuordnung zum Curriculum in diesem Studiengang:	M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	<p>Modul Compilerbau ist notwendige Voraussetzung, Java-Kenntnisse werden erwartet.</p> <p>Die Teilnehmerzahl in diesem Modul ist auf maximal 15 beschränkt.</p>		
12. Lernziele:	<p>Die Studierenden haben praktische Erfahrungen mit der Konstruktion eines Compilers und der Umsetzung von Konzepten in Programmiersprachen erworben. Sie sind in der Lage aktuelle Entwicklungen im Bereich der Programmiersprachen und des Compilerbaus zu beurteilen. Durch die Teilnahme an Programmierübungen mit Codereviews haben sie gelernt, qualitativ hochwertige Compiler zu entwickeln.</p>		
13. Inhalt:	<p>Lexer- und Parsergeneratoren, Semantische Attributierung, Fehlererkennung und -behandlung in Compilern, Typsysteme und Typprüfung, Die Java Virtual Machine, Zwischencodegenerierung, Sprachinterfaces</p>		
14. Literatur:	<p>A.W. Appel : Modern Compiler Implementation in Java 2nd Edition; Cambridge University Press (2002)</p> <p>A. V. Aho, M.S. Lahm, R. Sethi, J. D. Ullman: Compilers - Principles, Techniques, and Tools; Addison, Wesley (2007)</p>		
15. Lehrveranstaltungen und -formen:	581901 Vorlesung Entwurf und Implementierung eines Compilers		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden</p> <p>Selbststudium: 138 Stunden</p>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 58191 Entwurf und Implementierung eines Compilers (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0 • 58192 Entwurf und Implementierung eines Compilers (PL), schriftliche Prüfung, 90 Min., Gewichtung: 1.0 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 56980 Fachpraktikum Cloud Architekturen und Management

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

Modul: 29740 Fachpraktikum Eingebettete Systeme

2. Modulkürzel:	051711135	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	6.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	wiss. MA		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Modul "Embedded Systems Engineering"		
12. Lernziele:	Ability to apply the design methodology and commercial design tools for constructing and analyzing embedded hardware / software systems. Practical experience in software programming and debugging, digital circuit design and verification, usage of lab equipment such as logic analyzers. Experience in preparing structured technical documentation of specifications and designs.		
13. Inhalt:	This lab course focuses on analysis, design and implementation of embedded hardware/software systems and issues involved in the development of such systems. <ol style="list-style-type: none"> 1. Embedded software development 2. Usage of drivers for peripheral components 3. Cross-compilation 4. Remote debugging 5. Software performance profiling 6. Design of accelerator hardware digital circuits 7. Digital circuit simulation 8. FPGA implementation (synthesis) of digital circuits 9. Hardware / software interfacing 10. Integrated functional verification of hardware and software 		
14. Literatur:	Lab handouts Documentation of development tools (provided in the lab)		
15. Lehrveranstaltungen und -formen:	297401 Übung Fachpraktikum Eingebettete Systeme		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit im Labor: 64 Stunden Vor- und Nachbereitung: 116 Stunden Summe: 180 Stunden		
17. Prüfungsnummer/n und -name:	29741 Fachpraktikum Eingebettete Systeme (LBP), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)		

Modul: 24900 Fachpraktikum Rechnergestützte Methoden der Produktentwicklung

2. Modulkürzel:	051400006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Dieter Roller		
9. Dozenten:	<ul style="list-style-type: none"> • Dieter Roller • Julian Eichhoff 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Grundkenntnisse wie sie in „CAD und Produktmodelle“ vermittelt werden.		
12. Lernziele:	Die Studierenden sind nach erfolgreichem Besuch des Moduls in der Lage <ul style="list-style-type: none"> • typische Problemstellungen, Methoden und Technologien im Anwendungsbereich des jeweils behandelten CAx-Feldes sowie dessen Eingliederung in die Produktentwicklungskette unter Verwendung von Fachvokabular zu beschreiben • die vorgestellten Methoden und Technologien gegenüberzustellen und ihren Einsatz im Bezug zu vorgegebenen Problemstellungen zu begründen • die Funktionen eines Technologievertreters aus dem CAx-Feld bei einer vorgegebenen Problemstellung effektiv anwenden können 		
13. Inhalt:	Jedes Semester wechselnd werden zu einem konkreten CAx-Bereich (CAD, CAM, CAP, CAQ) folgende Inhalte behandelt: <ul style="list-style-type: none"> • Eingliederung des Unternehmensbereichs hinsichtlich Produktentstehung/Produktlebenszyklus; Funktionen, Prozesse und Tätigkeiten des Unternehmensbereichs; Informationsflüsse (zugrundeliegende Daten, Objekte und Artefakte) und weitere Verbindung zu anderen Unternehmensbereichen • Überblick zu typischen Problemstellungen und darauf anwendbare Methoden im Aufgabenfeld des Unternehmensbereichs; Methodenvergleich • Überblick der marktdominierenden CAx-Technologien zur Unterstützung des Aufgabenfeldes; Technologievergleich • Anwendungsbezogene Detailvorstellung der Funktionen und Repräsentationsformen eines Vertreters dieser Technologien im Rahmen konkreter Beispielprobleme 		
14. Literatur:	<ul style="list-style-type: none"> • D. Roller. CAD: Effiziente Anpassungs- und Variantenkonstruktion. Springer, Heidelberg, 1995. • S. Vajna, C. Weber, H. Bley, K. Zeman. CAx für Ingenieure - Eine praxisbezogene Einführung. Springer, Heidelberg, 2009. • G. Pahl, W. Beitz, J. Feldhusen, K.-H. Grote. Konstruktionslehre - Grundlagen erfolgreicher Produktentwicklung Methoden und Anwendung. Springer, Heidelberg, 2007. 		

15. Lehrveranstaltungen und -formen:	249001 Übung Fachpraktikum Rechnergestützte Methoden der Produktentwicklung
<hr/>	
16. Abschätzung Arbeitsaufwand:	
<hr/>	
17. Prüfungsnummer/n und -name:	24901 Fachpraktikum Rechnergestützte Methoden der Produktentwicklung (LBP), schriftlich oder mündlich, Gewichtung: 1.0
<hr/>	
18. Grundlage für ... :	
<hr/>	
19. Medienform:	
<hr/>	
20. Angeboten von:	
<hr/>	

Modul: 45750 Fachpraktikum Verteilte Systeme

2. Modulkürzel:	051200111	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	Frank Dürr		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Verteilte Systeme, Rechnernetze II		
12. Lernziele:	Die Teilnehmer besitzen die Fähigkeit, verteilte Anwendungen und Dienste zu entwerfen und zu implementieren. Sie besitzen praktische Kenntnisse in der Netzprogrammierung und der Programmierung von Client/Server-Anwendungen. Sie verfügen über praktische Kenntnisse über Technologien und Werkzeugen zur Implementierung und zum Testen verteilter Systeme.		
13. Inhalt:	<ul style="list-style-type: none"> - Socket-Programmierung - Höherwertige Kommunikationsprotokolle und Webservice- Schnittstellen (HTTP & XML/JSON, RPC, SOAP, REST) - Client/Server-Systeme - Peer-to-Peer- und Ad-hoc-Kommunikation - Entwicklungsumgebungen - Test verteilter Systeme 		
14. Literatur:	- A.S. Tanenbaum: Computer Networks, 4th Edition, 2003		
15. Lehrveranstaltungen und -formen:	457501 Fachpraktikum Verteilte Systeme		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Gesamt: 180 Stunden		
17. Prüfungsnummer/n und -name:	45751 Fachpraktikum Verteilte Systeme (PL), Sonstiges, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Verteilte Systeme		

Modul: 29440 Geometric Modeling and Computer Animation

2. Modulkürzel:	051900010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> • Daniel Weiskopf • Thomas Ertl • Guido Reina 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	<p>Basic computer graphics, for example:</p> <ul style="list-style-type: none"> • 10060 Computergraphik 		
12. Lernziele:	<p>Students gain an understanding of the fundamental concepts and techniques of geometric modeling and computer animation. This includes theoretical and mathematical foundations, important algorithms, and implementation aspects as well as practical experience with modeling and animation tools such as Maya.</p>		
13. Inhalt:	<p>This course covers foundations and methods for the modeling of scenes and for computer animation. This includes the representation of curves and surfaces, which are used by modeling and animation software for modeling of objects, description of the dynamics of parameters, or keyframe animation. Physically based animation describes motion via kinematic and dynamics laws of mechanics. Applications thereof include particle systems all the way to character animation and deformation.</p> <p>In particular, the following topics are covered:</p> <ul style="list-style-type: none"> • Description and modeling of curves: differential geometry of curves, polynomial curves in general, interpolation, Bezier curves, B-splines, rational curves, NURBS • Description and modeling of surfaces: differential geometry of surfaces, tensor product surfaces, Bezier patches, NURBS, ruled surfaces, Coons pathes • Subdivision schemes: basic concept, convergence and limit process, subdivision curves, subdivision surfaces • Overview of animation techniques • Keyframe animation, inverse kinematics • Physically based animation of points and rigid bodies: kinematics and dynamics • Particle systems: Reeves, flocking and boids, agent-based simulation • Cloth animation: continuum mechanics, mass-spring model, numerical solvers for ordinary differential equations, explicit and implicit integrators 		

- Collision: efficient collision detection, bounding volume hierachies, hierarchical space partitioning, collision handling, sliding and resting contact
- Fluid simulation: wave equation, Navier Stokes, level sets, particle level sets
- Basics of film production: camera, lighting, production process, storyboard

14. Literatur:	<ul style="list-style-type: none"> • D. Eberly, 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics. Morgan Kaufmann, 2000 • G. Farin: Curves and Surfaces for CAGD: A Practical Guide. Morgan Kaufmann, 2002 • R. Parent: Computer Animation: Algorithms and Techniques. Morgan Kaufmann, 2002 • W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling: Numerical Recipies - The Art of Scientific Computing. Cambridge University Press, 1986
15. Lehrveranstaltungen und -formen:	294401 Vorlesung mit Übungen Geometrische Modellierung und Animation
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden, Selbststudium: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 29441 Geometric Modeling and Computer Animation (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0, • V Vorleistung (USL-V), schriftlich, eventuell mündlich, Erfolgreiche Teilnahme an Übungen / exercises passed
18. Grundlage für ... :	
19. Medienform:	Video projector, blackboard, exercises using PCs
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme

Modul: 29450 Graphentheorie

2. Modulkürzel:	050420105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	<ul style="list-style-type: none"> • Ulrich Hertrampf • Volker Diekert • Manfred Kufleitner 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Grundvorlesungen in theoretischer Informatik		
12. Lernziele:	Die Studierenden beherrschen typische Denk- und Herangehensweisen aus der Graphentheorie. Die Beziehung zwischen diversen Graphparametern werden verstanden, ebenso wie ihre algorithmische Relevanz. Die Eigenschaften der wichtigsten Graphklassen erschließen sich den Studierenden.		
13. Inhalt:	Die Vorlesung behandelt algorithmische Problem und strukturelle Zusammenhänge bei Graphen. Im Einzelnen werden die folgenden Themen behandelt: <ul style="list-style-type: none"> • Eulergraphen • Cographen • Bipartite Graphen • Planare Graphen, Eulerformel, Satz von Kuratowski • Graphparameter • Perfekte Graphen • Graphenfärbungen und der Satz von Ramsey • Extremale Graphentheorie 		
14. Literatur:	<ul style="list-style-type: none"> • Reinhard Diestel: Graphentheorie. Springer, 2010. • Martin Aigner, Günter M. Ziegler: Das BUCH der Beweise. Springer, 2009. • Jacobus H. van Lint, Richard M. Wilson: A Course in Combinatorics. Cambridge University Press, 2nd edition, 2001. 		
15. Lehrveranstaltungen und -formen:	294501 Vorlesung mit Übungen Graphentheorie		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Gesamt: 180 h		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 29451 Graphentheorie (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Theoretische Informatik		

Modul: 29610 Hardware Based Fault Tolerance

2. Modulkürzel:	051710023	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> • Hans-Joachim Wunderlich • Michael Kochte 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	10140 Grundlagen der Rechnerarchitektur / Advanced Processor Architecture 10310 Rechnerorganisation		
12. Lernziele:	Knowledge of methods for reliability assessment of circuits and systems Knowledge of the main techniques for implementing fault tolerance Knowledge how to design fault tolerant circuits and systems		
13. Inhalt:	Micro- and Nano-electronic systems can exhibit failures both right after production and during their operation. Systems for which safety and security is of concern have to be designed in a way that the desired function can be delivered even if some components fail or produce erroneous outputs. This lecture presents the most important design techniques that allow to tolerate hardware faults up to a certain degree. The topics of the lecture are as follows: Terminology Measures of fault tolerance Techniques for structural and time redundancy Error detection and diagnosis Fault masking, repair, reconfiguration Fault-tolerant distributed systems		
14. Literatur:	Apart from lecture slides, the following books can be used to deepen on the topics of the lecture: I. Koren and C. M. Krishna: Fault-Tolerant Systems Morgan-Kaufman, 2007 P. K. Lala: Self-Checking and Fault-Tolerant Digital Design, Morgan Kaufmann Publishers (2001) D.K. Pradhan: Fault-Tolerant Computer Design, Prentice Hall (1996) R.N. Rao: E. Fujiwara, Error Control Coding for Computer Systems, Prentice Hall (1989) M.L. Bushnell: V.D. Agrawal, Essentials of Electronic Testing, Klumer Academic Publishers (2000)		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 296101 Vorlesung Hardware Based Fault Tolerance • 296102 Übung Hardware Based Fault Tolerance 		
16. Abschätzung Arbeitsaufwand:	Presence Time: 42 Stunden		

Self Study: 138 Stunden

Sum: 180 Stunden

17. Prüfungsnummer/n und -name: 29611 Hardware Based Fault Tolerance (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Writen exam 90 min or Oral exam 30 min

18. Grundlage für ... :

19. Medienform: Laptop presentation

20. Angeboten von: Institut für Technische Informatik

Modul: 42920 Hardware-Software-Codesign

2. Modulkürzel:	051711110	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Bachelor-Veranstaltung "Grundlagen der Eingebetteten Systeme" oder gleichwertige Kenntnisse		
12. Lernziele:	Ability to conceptualize systems so that an application-specific, optimized trade-off between hardware and software implementation of system functionality is achieved.		
13. Inhalt:	This module deals with the joint design and optimization of hardware and software for pre-defined applications, covering the following topics: 1. Models for system specification 2. Modelling and simulation with the SystemC library 3. Synthesis of system architectures 4. Resource allocation and operation binding 5. Partitioning of functionality among hardware and software 6. Scheduling and schedulability for parallel multi-core architectures 7. Methods for system optimization 8. Application specific instruction set processors (ASIPs) 9. Network-on-Chip (NoC) interconnect architectures		
14. Literatur:	J. Teich, Digitale Hardware/Software-Systeme, 2. Auflage, 2007		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 429201 Vorlesung Hardware-Software-Codesign • 429202 Übung Hardware-Software-Codesign 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	42921 Hardware-Software-Codesign (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Eingebettete Systeme (Embedded Systems Engineering)		

Modul: 42420 High Performance Computing

2. Modulkürzel:	051240040	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dirk Pflüger		
9. Dozenten:	<ul style="list-style-type: none"> • Martin Bernreuther • Dirk Pflüger • Miriam Mehl 		
10. Zuordnung zum Curriculum in diesem Studiengang:	M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	080300100 Mathematik für Informatiker und Softwaretechniker und 051240005 Numerische und Stochastische Grundlagen der Informatik bzw. 051240006 Einführung in die Numerik und Stochastik für Softwaretechniker		
12. Lernziele:	Fähigkeit, parallele Algorithmen auf unterschiedlichen parallelen Plattformen mit Hilfe geeigneter algorithmischer Modelle zu bewerten. Kenntnis verschiedener Programmiermodelle für Parallelrechner mit verteiltem und gemeinsamem Speicher. Fähigkeit, auch fortgeschrittene Implementierungsaufgaben aus dem Bereich des Höchstleistungsrechnens auf Basis ausgewählter Programmiermodelle zu bewältigen.		
13. Inhalt:	Die Vorlesung beschäftigt sich mit den Grundlagen paralleler Programmierung und paralleler Algorithmen speziell im Hinblick auf die Anwendungsbereiche Wissenschaftliches Rechnen und High Performance Computing. Verwandte Fragestellungen aus dem Bereich der Theorie (parallele Modelle und parallele Komplexität, etc.) sowie aus der Rechnertechnik (parallele Architekturen) werden begleitend diskutiert. Nach einer allgemeinen Einführung (Klassifizierung von Parallelrechnern, Ebenen von Parallelität, Performance und Architekturen, etc.), werden die Grundlagen paralleler Programme eingeführt (Notation/Syntax, Synchronisation und Kommunikation, Design paralleler Programme, etc.). Sowohl die Programmierung auf Systemen mit gemeinsamem Speicher als auch auf Systemen mit verteiltem Speicher werden besprochen. Dabei wird jeweils mindestens ein geeignetes Programmiermodell (z.B. OpenMP, MPI, CUDA) vertieft behandelt. Aus dem Bereich des High Performance Computing werden begleitend klassische Algorithmen und Implementierungstechniken als Beispiele behandelt, z.B. parallele Algorithmen aus der linearen Algebra (Matrixmultiplikation, etc. oder einfache Verfahren zur Lösung partieller Differentialgleichungen). Zusätzlich können Themen wie Lastverteilung und Lastbalancierung (Grundlagen, Algorithmen zur Partitionierung und Lastbalancierung, etc.) vorgestellt werden.		
14. Literatur:	<ul style="list-style-type: none"> • T. Rauber, G. Rüniger: „Parallele Programmierung“, 2. Aufl., Springer 2007; (in English: T. Rauber, G. Rüniger: „Parallel Programming: for Multicore and Cluster Systems“, Springer 2010) • K.A. Berman, J.L. Paul: "Sequential and Parallel Algorithms", PWS Publishing Company, 1997 		

	<ul style="list-style-type: none">• B. Chapman, G. Jost, R. van der Pas: "Using OpenMP - Portable Shared Memory Parallel Programming", MIT Press, 2008• W. Gropp, E. Lusk, und R. Thakur: "Using MPI-2: Advanced Features of the Message-Passing Interface", das Buch ist auch in deutscher Übersetzung erhältlich.• D. Kirk, W.-M. Hwu Programming Massively Parallel Processors
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 424201 Vorlesung High Performance Computing• 424202 Übung High Performance Computing
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudiumszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	42421 High Performance Computing (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Simulation großer Systeme

Modul: 51720 IT-Strategy

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Sven Lorenz		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	<p>Die Vorlesung fokussiert auf Management Strategien. Es wird erläutert, wie solche Strategien entwickelt und evaluiert werden.</p> <p>Teilnehmer der Vorlesung verstehen die Bestandteile einer IT Strategie. Sie können eine IT Strategie ableiten und entwickeln, basierend auf dem aktuellen Status eines Unternehmens. Insbesondere wird verstanden, was unter den Begriffen und Konzepten IT Organisation, Sourcing Management, Architektur Management, Qualitäts- und Risk-Management und schliesslich IT Landschaften zu verstehen ist und wie man damit umgeht.</p>		
13. Inhalt:	<p>Über die Einstiegsfragestellung „Was ist ‚Strategie‘?“ wird erläutert, was eine Unternehmensstrategie und eine IT-Strategie ist, wobei sowohl die klassischen Ansätze als auch neue Sichtweisen vorgestellt werden. Im Schwerpunkt „Strategieentwicklung“ wird auf die Ableitung der IT-Strategie aus der Unternehmensstrategie eingegangen. Ein kanonisches Vorgehensmodell wird eingeführt und anhand von Unternehmensbeispielen illustriert. Der Schwerpunkt „IT-Strategie als Prozess“ beginnt mit der Einbettung der IT-Strategieaufgaben in die bekannten IT Prozessmodelle wie ITIL und CobiT. Im Rahmen eines verallgemeinerten IT-Prozessmodells werden die einzelnen IT-Strategieprozesse (IT-Organisationsentwicklung, IT-Sourcing-Strategie, IT-Architektur-Management, IT-Bebauungsplanung, IT-Qualitätsmanagement und IT-Risikomanagement) in der Folge detailliert erläutert. Dabei werden klassische und State-of-the-art Methoden und Werkzeuge zur Unterstützung der IT-Strategieprozesse vorgestellt. Exkurse in das IT-Portfoliomanagement und in IT-Kennzahlensysteme runden die Vorlesungsinhalte ab.</p>		
14. Literatur:	<ul style="list-style-type: none"> • Helmut Krcmar, „Informationsmanagement“, Springer, 2010 • Jürgen Hofmann, Werner Schmitt, „Masterkurs IT-Management“, VIEWEG+TEUBNER, 2010W. • Brenner, A. Resch, V. Schulz, „Die Zukunft der IT in Unternehmen“, FAZ Buch, 2010 • Martin Kütz, „Kennzahlen in der IT“, dpunkt-Verlag, 2007 		
15. Lehrveranstaltungen und -formen:	517201 Vorlesung mit Übungen IT-Strategie		
16. Abschätzung Arbeitsaufwand:	Lecture & exercises: 42 hours		

Self-study: 138 hours

17. Prüfungsnummer/n und -name: 51721 IT-Strategy (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Architektur von Anwendungssystemen

Modul: 48500 Image Synthesis

2. Modulkürzel:	051903654	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Martin Fuchs		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf • Martin Fuchs 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	Modul 051900002 Computergraphik		
12. Lernziele:	<p>The students know the theoretical foundations of image synthesis and have practical expertise in programming of rendering systems. They know several approaches and algorithms for three-dimensional computer graphics, both for real-time and physically accurate rendering.</p>		
13. Inhalt:	<p>The class covers physically based rendering techniques such as ray/path tracing and radiosity, computer graphics models for light transport and light/scene interaction, as well as numerical methods such as Monte Carlo integration and finite element methods which approximate solutions to the rendering equation. In addition, techniques which specifically employ modern graphics processing hardware are covered which approximate physically correct solutions in interactive application scenarios by means of rasterization and image-space rendering.</p> <p>Specifically, the class covers:</p> <ul style="list-style-type: none"> graphics hardware and rasterization APIs by example of OpenGL textures and procedural models shading and shadow computations in rasterization pipelines scene graphs, culling and level-of-detail approaches physically based rendering and photo-realistic image synthesis local shading and material models, especially the BRDF the rendering equation ray tracing and Monte-Carlo approaches global illumination simulation (especially by means of radiosity, distribution ray tracing and path tracing) 		
14. Literatur:	<p>Andrew S. Glassner, Principles of Digital Image Synthesis, 1995J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics: Principle and Practice, 1990M. Pharr, G. Humphreys, Physically Based Rendering, 2004</p>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 485001 Lecture Image Synthesis • 485002 Exercise Image Synthesis 		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden</p>		

-
17. Prüfungsnummer/n und -name:
- 48501 Image Synthesis (PL), schriftlich oder mündlich, 120 Min.,
Gewichtung: 1.0
 - V Vorleistung (USL-V), schriftlich, eventuell mündlich
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 55630 Information Visualization and Visual Analytics

2. Modulkürzel:	051900099	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> • Michael Burch • Thomas Ertl • Daniel Weiskopf 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	Basic Human Computer Interaction		
12. Lernziele:	Student gains expertise about fundamental concepts and techniques of information visualization and visual analytics. This includes algorithms and mathematical background, data structures and implementation aspects as well as practical experience with widely available visualization tools.		
13. Inhalt:	<p>Topics covered in this course:</p> <ul style="list-style-type: none"> - Perception and Cognition - Graphs and Networks - Hierarchies and Trees - Multi-dimensional and high-dimensional data visualization - Time series visualization - Visual Analytics - Software Visualization - Geospatial visualization 		
14. Literatur:	<ul style="list-style-type: none"> • Colin Ware. Visual Thinking for Design • Colin Ware. Information Visualization. Perception for Design • Edward Tufte. The Visual Display of Quantitative Information • Robert Spence. Design for Interaction • Jim Thomas. Illuminating the Path 		
15. Lehrveranstaltungen und -formen:	556301 Vorlesung und Übung Informationsvisualisierung		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden</p> <p>Gesamt: 180 Stunden</p>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 55631 Information Visualization and Visual Analytics (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0 		

-
- V Vorleistung (USL-V), schriftlich, eventuell mündlich, Erfolgreiche Übungsteilnahmen / excercises passed
-

18. Grundlage für ... :

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

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

Modul: 29630 Konzepte der Programmiersprachen

2. Modulkürzel:	051510312	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Kenntnisse die in etwa den Inhalten des Moduls 051510015 - Grundlagen des Compilerbaus und der Programmiersprachen - des Bachelor-Studiums entsprechen.		
12. Lernziele:	<p>Die Studierenden haben viele den Programmiersprachen zugrundeliegende Konzepte in ihren Variationen kennengelernt und verstanden. Sie haben unterschiedliche Ausführungsmodelle in ihren sprachlichen Ausprägungen kennengelernt. Diese Kenntnisse ermöglichen das schnellere Erlernen weiterer Sprachen und ein vertieftes Verständnis ihnen bekannter Sprachen sowie das Vermeiden von bekannten Fehlern beim Entwurf neuer Sprachen.</p> <p>(Das Modul wird in Reaktion auf die Entstehung und die 2012 erfolgte Erweiterung des Moduls "Programmierparadigmen" ab 2014 grundsätzlich überarbeitet oder gestrichen.)</p>		
13. Inhalt:	<p>Überblick typischer Konzepte in Programmiersprachen und ihrer Realisierung durch Übersetzer oder Interpreter. Bindungskonzepte, Datentypen und Typsysteme, Abstraktion und Kompositionsmechanismen, Konzepte objekt-orientierter Sprachen, sequentielle und parallele Kontrollkonstrukte, synchrone und asynchrone Kommunikationskonstrukte. Ausführungsmodelle für imperative, objekt-orientierte, funktionale und logische Programmiersprachen, sowie beispielhafte Sprachelemente.</p> <p>Die Vorlesung ist kein Streifzug durch diverse Programmiersprachen, sondern die Vorstellung zugrundeliegender Prinzipien, und ihrer Begründung aus der Sicht des Software Engineering und der Implementierungsmodelle.</p>		
14. Literatur:	<ul style="list-style-type: none"> • Ghezzi, Jazayeri, Programming Language Concepts, 1987 • Sebesta, Robert, Concepts of Programming Languages, 2003 		
15. Lehrveranstaltungen und -formen:	296301 Vorlesung mit Übung Konzepte der Programmiersprachen		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29631 Konzepte der Programmiersprachen (PL), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von: Institut für Softwaretechnologie

Modul: 29460 Kryptographische Verfahren

2. Modulkürzel:	050420110	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof. Ulrich Hertrampf		
9. Dozenten:	<ul style="list-style-type: none"> • Ulrich Hertrampf • Volker Diekert • Stefan Funke 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Theorie-Vorlesungen des Bachelor-Studiums		
12. Lernziele:	Die Studierenden kennen die wichtigsten Sätze der Kryptographie. Sie können klassische und moderne Verschlüsselungsverfahren anwenden und die Sicherheit dieser Verfahren beurteilen und einstufen.		
13. Inhalt:	Moderne Verfahren der einstigen "Geheimwissenschaft" Kryptographie werden eingeführt. Die Veranstaltung stellt Methoden zur Erzeugung elektronischer Unterschriften und zur Identifikation von Benutzern vor, die als notwendige Voraussetzungen für elektronische Wahlen oder anonymes elektronisches Bargeld gelten. Es werden neben klassischen, symmetrischen Verschlüsselungsverfahren aktuelle asymmetrische Verfahren behandelt. Eine wichtige Rolle spielen Protokolle, die aufbauend auf kryptographischen Verfahren die erwähnten Aufgaben lösen.		
14. Literatur:	<ul style="list-style-type: none"> • Bruce Schneier, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C, 1996 • Douglas Robert Stinson, Cryptography: Theory and Practice, 1995 • Friedrich Ludwig Bauer, Entzifferte Geheimnisse: Methoden und Maximen der Kryptologie, 1995 • Johannes Buchmann, Einführung in die Kryptographie, 1999 		
15. Lehrveranstaltungen und -formen:	294601 Vorlesung mit Übungen Kryptographische Verfahren		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 29461 Kryptographische Verfahren (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Theoretische Informatik		

Modul: 29480 Loose Coupling and Message Based Applications

2. Modulkürzel:	052010003	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> • Frank Leymann • Dimka Karastoyanova 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Understand the problem of application integration and the fundamental concept of loose coupling. The pros and cons of messaging are clear, and the architecture of Message Oriented Middleware is understood. Key patterns of using messaging to solve (enterprise) application integration problems are understood.		
13. Inhalt:	<p>Messaging is a cornerstone of the integration of heterogeneous applications inside and among enterprises. Applications that need to share data synchronously or asynchronously with each other can be made to interoperate by means of the feature-rich Message-Oriented Middleware (MOM) that has grown ubiquitous in enterprises. During this course we treat the approaches and challenges of application integration through messaging. At first, we will address concepts such as (a-)synchronous messaging and the different messaging styles, e.g. point-to-point and publish-subscribe, that are the foundation of message-based application integration. Later in the course we will take an in-depth look at the mechanics and architecture of MOM, in particular of the Java Messaging Service (JMS), which will also be used in examples and exercises. Throughout the course we will discuss and apply extensively Enterprise Application Integration (EAI) patterns. Especially, endpoint patterns, routing patterns, transformation patterns, messaging patterns, channel patterns, and management patterns will be presented; the composability of these patterns will be explained.</p>		
14. Literatur:	<p>G. Hohpe and B. Woolf: "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions." Addison-Wesley Professional, ISBN-13: 978-0321200686. October 2003.</p> <p>M. Hapner et al: "Java Messaging Service API Tutorial & Reference". Addison-Wesley 2001.</p>		
15. Lehrveranstaltungen und -formen:	294801 Vorlesung mit Übungen Lose Kopplung & Message-basierte Integration		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		

Modul: 29470 Machine Learning

2. Modulkürzel:	051200112	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	Marc Toussaint		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Solid knowledge in Linear Algebra, probability theory and optimization. Fluency in at least one programming language.		
12. Lernziele:	Students will acquire an in depth understanding of Machine Learning methods. The concepts and formalisms of Machine Learning are understood as generic approach to a variety of disciplines, including image processing, robotics, computational linguistics and software engineering. This course will enable students to formalize problems from such disciplines in terms of probabilistic models and the derive respective learning and inference algorithms.		
13. Inhalt:	<p>Exploiting large-scale data is a central challenge of our time. Machine Learning is the core discipline to address this challenge, aiming to extract useful models and structure from data. Studying Machine Learning is motivated in multiple ways: 1) as the basis of commercial data mining (Google, Amazon, Picasa, etc), 2) a core methodological tool for data analysis in all sciences (vision, linguistics, software engineering, but also biology, physics, neuroscience, etc) and finally, 3) as a core foundation of autonomous intelligent systems (which is my personal motivation for research in Machine Learning).</p> <p>This lecture introduces to modern methods in Machine Learning, including discriminative as well as probabilistic generative models. A preliminary outline of topics is:</p> <ul style="list-style-type: none"> • motivation and history • probabilistic modeling and inference • regression and classification methods (kernel methods, Gaussian Processes, Bayesian kernel logistic regression, relations) • discriminative learning (logistic regression, Conditional Random Fields) • feature selection • boosting and ensemble learning • representation learning and embedding (kernel PCA and derivatives, deep learning) • graphical models • inference in graphical models (MCMC, message passing, variational) • learning in graphical models • structure learning and model selection • relational learning <p>Please also refer to the course web page: http://ipvs.informatik.uni-stuttgart.de/mlr/marc/teaching/13-MachineLearning/</p>		

14. Literatur:	<p>[1] <i>The Elements of Statistical Learning: Data Mining, Inference, and Prediction</i> by Trevor Hastie, Robert Tibshirani and Jerome Friedman. Springer, Second Edition, 2009. full online version available: http://www-stat.stanford.edu/~tibs/ElemStatLearn/ (recommended: read introductory chapter)</p> <p>[2] <i>Pattern Recognition and Machine Learning</i> by Bishop, C. M.. Springer 2006. online: http://research.microsoft.com/en-us/um/people/cmbishop/prml/ (especially chapter 8, which is fully online)</p>
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 294701 Lecture Machine Learning • 294702 Exercise Machine Learning
16. Abschätzung Arbeitsaufwand:	<p>Presence time: 42 hours Self study: 138 hours Sum: 180 hours</p>
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 29471 Machine Learning (PL), schriftlich, eventuell mündlich, 180 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Parallele und Verteilte Systeme

Modul: 29640 Mikrocontroller

2. Modulkürzel:	051230115	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Knowledge of at least one programming language and knowledge in the field of computer science or similar subjects. Kenntnisse in mindestens einer Programmiersprache und in mindestens einem Fach aus dem Bereich dem Bereich der Technischen Informatik oder ähnlichen Fächern.		
12. Lernziele:	Students are able to master the practical programming of microcontrollers and are familiar with classical architectures. Historical Overview Microcontroller architectures Applications of microcontrollers Instruction set classic microcontroller Assembly language programming of microcontrollers C programming for microcontrollers Studierende beherrschen die praktische Programmierung von Mikrocontrollern und kennen klassische Architekturen. <ul style="list-style-type: none"> • Historische Übersicht • Mikrocontroller-Architekturen • Einsatzgebiete von Mikrocontrollern • Befehlssatz klassischer Microcontroller • Assembler-Programmierung von Mikrocontrollern • C-Programmierung von Mikrocontrollern 		
13. Inhalt:	Microcontrollers (also called μ Controller, μ C, MCU) are IC's that combine at least peripheral functions on a single chip. In many cases, the working and programming memory is also partially or completely on the same chip. A microcontroller is practically a one-chip computer system. The number of built-in microcontroller exceeds by far the number of microprocessors. A microcontroller is often part of an embedded system in devices of everyday life like washing machines, smart cards (money, telephone cards), consumer electronics (VCRs, disc players, radios, televisions, remote controls), office electronics, motor vehicles (ECU for ABS, airbag, engine, instrument cluster, ESP, etc.), mobile phones and even in clocks and watches. In addition they are found on virtually all computer peripherals including keyboards, mouse, printers, monitors, scanners etc.		

Microcontrollers are adapted to performance and respective features of the application. Therefore they have significant advantages in cost and power consumption compared with normal computers. Small microcontrollers are available in high numbers for less than 1\$.

Als Microcontroller (auch μ Controller, μ C, MCU) werden ICs bezeichnet, die mit dem Prozessor mindestens Peripheriefunktionen auf einem Chip vereinen. In vielen Fällen befindet sich der Arbeits- und Programmspeicher ebenfalls teilweise oder komplett auf dem gleichen Chip. Ein Mikrocontroller ist praktisch ein Ein-Chip-Computersystem. Die Anzahl der verbauten Mikrocontroller überschreitet bei weitem die Zahl der Mikroprozessoren.

Der Mikrocontroller tritt in Gestalt von eingebetteten Systemen im Alltag oft unbemerkt in technischen Gebrauchsartikeln auf, zum Beispiel in Waschmaschinen, Chipkarten (Geld-, Telefonkarten), Unterhaltungselektronik (Videorekordern, CD-/DVD-Playern, Radios, Fernsehgeräten, Fernbedienungen), Büroelektronik, Kraftfahrzeugen (Steuergeräte für z.B. ABS, Airbag, Motor, Kombiinstrument, ESP usw.), Mobiltelefonen und sogar in Uhren und Armbanduhren. Darüber hinaus sind sie in praktisch allen Computer-Peripheriegeräten enthalten (Tastatur, Maus, Drucker, Monitor, Scanner uvm.).

Mikrocontroller sind in Leistung und Ausstattung auf die jeweilige Anwendung angepasst. Daher haben sie gegenüber normalen Computern deutliche Vorteile bei den Kosten und der Leistungsaufnahme. Kleine Mikrocontroller sind in höheren Stückzahlen für deutlich unter 1\$, - verfügbar.

Aus <http://de.wikipedia.org/wiki/Mikrocontroller>

14. Literatur:

- Jörg Wiegelmann, Softwareentwicklung in C für Mikroprozessoren und Mikrocontroller: C- Programmierung für Embedded-Systeme, 2009

More literature is named in the lecture

15. Lehrveranstaltungen und -formen: 296401 Vorlesung mit Übung Mikrocontroller

16. Abschätzung Arbeitsaufwand: Präsenzzeit: 42 Stunden
Selbststudium: 138 Stunden
Gesamt: 180 Stunden

17. Prüfungsnummer/n und -name: 29641 Mikrocontroller (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündlichen Prüfung von 30 Min.

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 29720 Mobile Computing

2. Modulkürzel:	051200166	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> • Kurt Rothermel • Frank Dürr 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Rechnernetze		
12. Lernziele:	<p>The knowledge that has been acquired in the course "Computer Networks I" regarding concepts, protocols, and technologies of computer networks, will be extended to mobile devices and wireless communication systems and procedures. The objective of this lecture is to understand problems that might occur in the usage of mobile devices as well as to obtain knowledge to develop solutions for these problems and to communicate with experts. The Participants will learn about advantages and the disadvantages of specific wireless communication technologies for mobile devices and will be able to use appropriate protocols for the applications or modify them as needed. The exercises are used to provide practical experience in programming, analysis, performance evaluation of mobile and wireless communication systems as well as the expertise in the usage of appropriate tools.</p>		
13. Inhalt:	<ol style="list-style-type: none"> 1. Fundamentals of wireless data transmission 2. Media access for wireless networks 3. Location Management 4. Wireless Wide Area Networks 5. Wireless networks (local/personal) 6. Ad-hoc Networks: Exchange, Location administration 7. Mobility in IP-networks 8. Transport layer protocols for mobile systems 9. Location of services 10. Mobile data access 11. Introduction 12. Wireless data transmission 13. Location Management 14. Wireless 15. Telephone communication systems : GSM, GPRS,UMTS 16. Wireless networks (local/personal): 802.11, Bluetooth 17. Ad-hoc Networks: Routing, Location Management 18. Internetworking: Mobile IP, Cellular IP 19. Transport layers for mobile systems 20. Location of services : Problem, JINI, UpnP 21. Mobile data access: Broadcast Scheduling, Hoarding 		
14. Literatur:	Charles E. Perkins: Mobile IP: Design Principles and Practices. 1997 James D. Solomon: Mobile IP: The Internet Unplugged. 1998 Jochen Schiller: Mobile Communications. 2000 Jörg Roth: Mobile Computing: Grundlagen, Technik und Konzepte. 2002		

Kian-Lee Tan, Beng-Chin Ooi: Data Dissemination in Wireless Computing Envi-ronments. 2000
Tomasz Imielinski, Henry F. Korth (ed.): Mobile Computing. 1996

15. Lehrveranstaltungen und -formen:	297201 Vorlesung mit Übung Mobile Computing
16. Abschätzung Arbeitsaufwand:	Mobile Computing Vorlesung - Präsenzzeit: 21 Stunden - Selbststudium: 69 Stunden Mobile Computing Übungen - Präsenzzeit: 21 Stunden - Selbststudium: 69 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 29721 Mobile Computing (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0• V Vorleistung (USL-V), schriftlich, eventuell mündlich• V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	Folien, Tafel
20. Angeboten von:	Verteilte Systeme

Modul: 29730 Modelling, Simulation, and Specification

2. Modulkürzel:	051711020	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Martin Radetzki		
9. Dozenten:	Martin Radetzki		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	Master-level understanding of and practical experience with fundamental models of computation and their simulation, ability to apply them to embedded systems specification.		
13. Inhalt:	<p>Given the complexity and implementation cost of contemporary electronic systems, it is essential to specify their intended functionality before elaborating the implementation. This course focuses on the model-based and executable specification of embedded systems and covers the following topics:</p> <ul style="list-style-type: none"> • Hierarchical concurrent state machine models; • Kahn process networks, synchronous data flow networks; • Specification of timing, concurrency, and non-functional aspects; • Object-oriented modelling of embedded systems; • Event-driven simulation; • Modelling levels with emphasis on transaction level modelling; • Application to embedded systems specification; • SystemC. 		
14. Literatur:	<ul style="list-style-type: none"> • Lecture Notes “Modelling, Simulation, and Specification”. • Jantsch: Modeling Embedded Systems and SoCs Concurrency and Time in Models of Computation. Morgan Kaufman Publishers, 2004. • Black, D.; Donovan, D.: SystemC from the Ground Up. Kluwer Academic Publishers, 2004. 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 297301 Vorlesung Modelling, Simulation, and Specification • 297302 Übung Modelling, Simulation, and Specification 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden Summe: 180 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 29731 Modelling, Simulation, and Specification (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich, Als Zulassungsvoraussetzung zur Klausur ist die folgende 		

Vorleistung zu erbringen: Teilnahme an den Übungen,
Präsentation der Lösung wenigstens einer Aufgabe.

18. Grundlage für ... :

19. Medienform:

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

Modul: 55650 Multimodal Interaction for Ubiquitous Computers

2. Modulkürzel:	051900033	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Albrecht Schmidt		
9. Dozenten:	<ul style="list-style-type: none"> • Albrecht Schmidt • Niels Henze 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Basics of human computer interaction		
12. Lernziele:	Broad understanding for methods and concepts of multimodal interactions of personal computers, in particular for mobile systems, vehicles, tedious devices and environments.		
13. Inhalt:	<ul style="list-style-type: none"> • Interaction with mobile phones • User interfaces for vehicles • Interaction with intelligent environments • Interactive interfaces and gestures • Tangible user interfaces • Speech input and output • Camera-based interaction • Physiological sensors as interfaces between human and computer • Activities, context and emotions as input • Methods and techniques for designing user interfaces • Approaches for evaluating user interfaces 		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 556501 Lecture Multimodal Interaction for Ubiquitous Computers • 556502 Exercise Multimodal Interaction for Ubiquitous Computers 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	55651 Multimodal Interaction for Ubiquitous Computers (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme		

Modul: 40680 Optimization

2. Modulkürzel:	051200113	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	Marc Toussaint		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Solid basic knowledge in linear algebra and analysis. Basic programming skills.		
12. Lernziele:	<p>Students will learn to identify, mathematically formalize, and derive algorithmic solutions to optimization problems as they occur in nearly all disciplines, e.g. Machine Learning, Combinatorial Optimization, Computer Vision, Robotics, Simulation. The focus will be on continuous optimization problems (including as they arise from relaxations of discrete problems), including convex problems, quadratic & linear programming, but also non-linear black-box problems. The goal is to give an overview of the various approaches and mathematical formulations and practical experience with the basic paradigms.</p>		
13. Inhalt:	<p>Optimization is one of the most fundamental tools of modern sciences. Many phenomena -- be it in computer science, artificial intelligence, logistics, physics, finance, or even psychology and neuroscience -- are typically described in terms of optimality principles. The reason is that it is often easier to describe or design an optimality principle or cost function rather than the system itself. However, if systems are described in terms of optimality principles, the computational problem of optimization becomes central to all these sciences.</p> <p>This lecture aims give an overview and introduction to various approaches to optimization together with practical experience in the exercises. The focus will be on continuous optimization problems and we will cover methods ranging from standard convex optimization and gradient methods to non-linear black box problems (evolutionary algorithms) and optimal global optimization. Students will learn to identify, mathematically formalize, and derive algorithmic solutions to optimization problems as they occur in nearly all disciplines. A preliminary list of topics is:</p> <ul style="list-style-type: none"> • gradient methods, log-barrier, conjugate gradients, Rprop • constraints, KKT, primal/dual • Linear Programming, simplex algorithm • (sequential) Quadratic Programming • Markov Chain Monte Carlo methods • 2nd order methods, (Gauss-)Newton, (L)BFGS • blackbox stochastic search, including a discussion of evolutionary algorithms <p>Please also refer to the course web page: http://ipvs.informatik.uni-stuttgart.de/mlr/marc/teaching/13-Optimization/</p>		

14. Literatur:

15. Lehrveranstaltungen und -formen: 406801 Vorlesung mit Übungen Optimization

16. Abschätzung Arbeitsaufwand: Presence time: 42 hours
Self study: 138 hours
Sum: 180 hours

17. Prüfungsnummer/n und -name: 40681 Optimization (PL), schriftlich, eventuell mündlich, 120 Min.,
Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Parallele und Verteilte Systeme

Modul: 56790 Parallele Numerik

2. Modulkürzel:	051240080	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Miriam Mehl		
9. Dozenten:	<ul style="list-style-type: none"> • Miriam Mehl • Dirk Pflüger • Stefan Zimmer 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	<p>Modul 41590 Einführung in die Numerik und Stochastik für Softwaretechniker oder</p> <p>Modul 10240 Numerische und Stochastische Grundlagen</p>		
12. Lernziele:	<p>Die Studenten kennen die wesentlichen parallelisierbaren Algorithmen für zentrale numerische Problemstellungen. Sie erkennen Parallelisierungshindernisse in bekannten und neuen numerischen Algorithmen, können die zu erwartende Skalierbarkeit abschätzen und sind in der Lage, Algorithmen so zu modifizieren, dass die parallele Effizienz erhöht wird ohne wichtige numerische Eigenschaften wie Stabilität und Komplexität zu verlieren.</p>		
13. Inhalt:	<ul style="list-style-type: none"> • parallele Matrix- und Vektoroperationen • parallele Fouriertransformation • parallele QR Zerlegung und Least Squares Probleme • parallele iterative Gleichungssystemlöser • parallele Eigenwert- und Eigenvektorberechnung • parallele Zeitschrittverfahren • parallele Algorithmen für Teilchenwechselwirkungen 		
14. Literatur:	<ul style="list-style-type: none"> • Introduction to High Performance Scientific Computing (Eijkhout, Chow, van de Geijn) (download at http://www.lulu.com/shop/victor-eijkhout/introduction-to-high-performance-scientific-computing/paperback/product-21431780.html;jsessionid=CF30CC0B65B0F349BFBD206D406F8) • Numerical Linear Algebra for High-Performance Computers (Dongarra, Duff, Sorensen, van der Vorst) • Parallel Algorithms for Matrix Computations (Gallivan, Heath, Ng, Ortega,...) • A User's Guide to MPI (Pacheco) • Iterative Methods for Sparse Linear Systems (Saad) • Loesung linearer Gleichungssysteme auf Parallelrechnern (Frommer) • M. Griebel, S. Knapek, G. Zumbusch, and A. Caglar. Numerische Simulation in der Molekulardynamik. Springer, 2004. • D. Frenkel and B. Smith. Understanding Molecular Simulation from Algorithms to Applications. Academic Press (2nd ed.), 2002. 		

Modul: 29650 Parallele Programmierung

2. Modulkürzel:	051230130	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Kenntnisse in mindestens einer Programmiersprache. Kenntnisse in mindestens einem Fach der Technischen Informatik odereinem ähnlichen Fach.		
12. Lernziele:	Studierende beherrschen die Programmierung von Multi-Core Prozessoren und paralleler Rechner bzw. Computing-Systemen.		
13. Inhalt:	<ul style="list-style-type: none"> • Grundlegende Parallelisierungsansätze: Parallelisierung durch Datenzerlegung, parallele lineare Algebra, etc. • Message Passing Interface • Open MP • C-Programmierung für FPGAs • Graphische Programmierung • GPU-Programmierung 		
14. Literatur:	<ul style="list-style-type: none"> • Thomas Rauber und Gundula Rürger, Multicore: Parallele Programmierung (Informatik Im Fokus), 2007 • More literature is named in the lecture 		
15. Lehrveranstaltungen und -formen:	296501 Vorlesung mit Übung Parallele Programmierung		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29651 Parallele Programmierung (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 10250 Parallele Systeme

2. Modulkürzel:	051200065	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Erfahrungen aus dem Bereich Technische Informatik		
12. Lernziele:	Grundlegende Kenntnisse im Bereich paralleler Systeme, z.B. Multi-Core CPUs und deren Programmierung.		
13. Inhalt:	<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:	Wird in der Lehrveranstaltung bekannt gegeben.		
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	
	Gesamt:	180 Stunden	
17. Prüfungsnummer/n und -name:	10251 Parallele Systeme (LBP), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 48570 Practical Course Visual Computing

2. Modulkürzel:	051900111	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	Thomas Ertl		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Basics of Computer Graphics		
12. Lernziele:	During this practical course, students will learn about approaches to rendering and visual computing technologies and will know how to implement these. They will learn about polygon based approach as well as volume rendering approaches. The students will learn, how to proceed a small project on their own (independently).		
13. Inhalt:	OpenGLQt-FrameworkRaytracingVolume RenderingIndependent Project		
14. Literatur:	<ul style="list-style-type: none"> • OpenGL Programming Guide - Third Edition (OpenGL 1.2) , Masonn Woo, Jackie Neider, Tom Davis, Dave Shreiner, Addison Wesley, 1999 • Programming with Qt - First Edition, Matthias Kalle Dalheimer, O'Reilly,1999 • An Introduction to Ray Tracing, Andrew S. Glassner, Academic Press, 1989 • Computer Graphics - Principle and Practice - Second Edition, Foley, van Dam, Feiner, Huges, Addison Wesley, 1990 		
15. Lehrveranstaltungen und -formen:	485701 Lab Practical Course Visual Computing		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	48571 Practical Course Visual Computing (LBP), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 29660 Programmanalysen und Compilerbau

2. Modulkürzel:	051510311	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Kenntnisse, die in etwa den Inhalten des Moduls 051510015 - Grundlagen des Compilerbaus und der Programmiersprachen - des Bachelor-Studiums entsprechen, sind dringend empfohlen.		
12. Lernziele:	Die Studierenden haben grundlegende Kenntnisse über die typischen in Compilern und verwandten Programmanalysen erworben, sowohl in Bezug auf Basisanalysen (Kontroll- und Datenflussanalysen) als auch auf weitergehende, zielgerichteten Analysen wie Zeigeranalysen, Abhängigkeitsanalysen oder Slicing. Speziell lernen sie eine Reihe von Codeoptimierungen im Compiler kennen, aber auch Globalanalysen, wie sie zur Fehlersuche, zum Reengineering oder zu Architekturanalysen nötig sind. Ferner erhalten sie eine Einführung in die Codegenerierung in Compilern.		
13. Inhalt:	<ul style="list-style-type: none"> • Attributgrammatiken (Wiederholung) • Speicherorganisation (Speicherverwaltung, Aktivierungsblöcke) • Zwischencode-Erzeugung • Programmanalysen und -Optimierung (Schwerpunkt) • klassische Optimierungen • Lokale und globale Kontrollflussanalyse • Lokale und globale Datenflussanalysen • Dominatoren, Dominatorgrenzen, Kontrollstrukturanalysen • Zeigeranalysen • Seiteneffekt-Analyse • Datenabhängigkeiten, Konfliktanalysen und Registervergabe • SSA-Form und ihre Berechnung • Code-Erzeugung • Implementierung von OOP • Das Laufzeitsystem • Separate Übersetzung 		
14. Literatur:	<ul style="list-style-type: none"> • Aho, Sethi, Ullman, Compilers - Principles, Techniques, and Tools, 1988 • Morgan, Robert, Building an Optimizing Compiler, 1998 • Muchnick, Steven S., Advanced Compiler Design and Implementation, 1997 		
15. Lehrveranstaltungen und -formen:	296601 Vorlesung mit Übung Programmanalysen und Compilerbau		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29661 Programmanalysen und Compilerbau (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Softwaretechnologie

Modul: 51740 Quantencomputing

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

Modul: 29670 Rapid Prototyping

2. Modulkürzel:	051230135	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Erfahrungen in mindestens einer Programmiersprache.		
12. Lernziele:	Die Studierenden beherrschen die schnelle Realisierung von Computing-Systemenausgehend von einer Algorithmen-Implementierung unter Verwendung eines Computer-Algebrasystems.		
13. Inhalt:			
14. Literatur:	<ul style="list-style-type: none"> • James O. Hamblen und Michael D. Furman, Rapid Prototyping of Digital Systems: A Tutorial Approach, 2001 • More literature is named in the lecture 		
15. Lehrveranstaltungen und -formen:	296701 Vorlesung mit Übung Rapid Prototyping		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29671 Rapid Prototyping (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 29680 Real-Time Programming

2. Modulkürzel:	051510301	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Erhard Plödereder		
9. Dozenten:	Erhard Plödereder		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Significant programming experience (not necessarily in real-time application) is highly advisable. Knowledge of Ada, C/C++ and Unix is helpful, but not required.		
12. Lernziele:	Students understand the standard terminology of deadline-driven, safety-critical real-time systems. They understand the issues that differentiate such systems from general software systems, and they know about available solutions, if any.		
13. Inhalt:	<ol style="list-style-type: none"> 1) General requirements and terminology of real-time systems 2) Deterministic execution: avoiding language-, implementation- and hardware-induced non-determinisms; coping with limited resources; storage estimation and management; execution time estimation 3) Fault tolerance: Faults and failure modes, N-version programming, voting, forward and backward recovery 4) Simple scheduling regimes: cyclic executives, deadline guarantees 5) Parallelism and priority scheduling regimes: processes, threads, tasks; run-time kernels; task management; interrupt handling 6) Synchronization and communication: semaphores, critical regions, monitors, protected objects, rendezvous, messaging 7) Control of shared resources 8) Distributed Systems: basic concepts; major issues 		
14. Literatur:	<ul style="list-style-type: none"> • Alan Burns and Andy Wellings, Real-Time Systems and Programming Languages, 1997 ... or later editions of the Burns/Wellings-Book, e.g., 4.ed. 2009 Language reference manuals (C++, Java, Ada) are useful at times.		
15. Lehrveranstaltungen und -formen:	296801 Vorlesung mit Übung Real-Time Programming		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29681 Real-Time Programming (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Softwaretechnologie		

Modul: 29690 Real-Time Video Processing I

2. Modulkürzel:	051230140	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	This course requires knowledge and experience in (at least) one programming language as well as knowledge of the subject "Technische Informatik" or a similar course		
12. Lernziele:	The Students will gain knowledge in the implementation of algorithms, architectures and exemplary processors for real-time video processing		
13. Inhalt:	<ul style="list-style-type: none"> • Introduction: analog/digital Television • Cameras, Image sensors and their characteristics • Image Filtering, Bayer Filter • Motion Analysis • video compression • video communication • video processing • Parallel architecture, video processors and Implementation of hardware components for real-time video processing algorithms 		
14. Literatur:	<ul style="list-style-type: none"> • Roger Clarke und R. J. Clarke von Academic Press Inc, Digital Compression of Still Images and Video (Signal Processing and Its Applications), 1995 • More literature is named in the lecture 		
15. Lehrveranstaltungen und -formen:	296901 Vorlesung mit Übung Real-Time Video Processing I		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29691 Real-Time Video Processing I (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 29700 Real-Time Video Processing II

2. Modulkürzel:	051230142	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankuendigung
8. Modulverantwortlicher:	Univ.-Prof. Sven Simon		
9. Dozenten:	Sven Simon		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Voraussetzung für Real-Time Video Processing II sind Kenntnisse von Real-Time Video Processing I. Alternativ sind Kenntnisse aus einem Fach der Technischen Informatik oder einem ähnlichen Fach oder Kenntnisse im Bereich der Datenkompression oder der Bildverarbeitung oder der Signalverarbeitung Voraussetzung.		
12. Lernziele:	Die Studierenden können praktisch Real-Time Video Processing Systeme aufbauen.		
13. Inhalt:	<ul style="list-style-type: none"> • Vorstellung der Fallstudie eines Video Processing Systems • Auswahl der Algorithmen des Video Processing Systems • Implementierung und Verifikation der Algorithmen • Architektur-Entwicklung des Video Processing Systems • Performance-Analyse der Achitektur • Implementierung und System-Verifikation 		
14. Literatur:	<ul style="list-style-type: none"> • Roger Clarke und R. J. Clarke von Academic Press Inc, Digital Compression of Still Images and Video (Signal Processing and Its Applications), 1995 • More literature is named in the lecture 		
15. Lehrveranstaltungen und -formen:	297001 Vorlesung mit Übung Real-Time Video Processing II		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	29701 Real-Time Video Processing II (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 45740 Rechnernetze II

2. Modulkürzel:	051200168	5. Moduldauer:	2 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> • Kurt Rothermel • Frank Dürr 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Rechnernetze		
12. Lernziele:	<p>Das in der Veranstaltung Rechnernetze I erworbene Verständnis von Konzepten, Protokollen und Technologien von Rechnernetzen wird weiter vertieft. Der Teilnehmer kennt die Funktionsweise der wichtigsten Dienste der Anwendungsschicht des Schichtenmodells. Er kennt die Konzepte zur Realisierung von Netzen auf Anwendungsebene. Er ist im Stande, diese Dienste und Konzepte bei der Konzeption eigener Anwendungen zu nutzen und ebenso eigene Dienste und Systeme zu entwickeln, um konkrete Probleme zu lösen.</p>		
13. Inhalt:	<p>Vorlesung Höhere Kommunikationskonzepte und -Protokolle:</p> <ol style="list-style-type: none"> 1. Einführung 2. Socket-Schnittstelle 3. Präsentation und Kompression 4. Realzeitkommunikation 5. Elektronische Bezahlsysteme 6. Multicast auf Anwendungsschicht 7. Inhaltsbezogene Netze 8. Geographische Kommunikation <p>Vorlesung Peer-to-Peer-Systeme:</p> <ol style="list-style-type: none"> 1. Definition und Anwendungen von Peer-to-Peer-Systemen 2. Theoretische Netzmodelle 3. Unstrukturierte Peer-to-Peer-Systeme 4. Strukturierte Peer-to-Peer-Systeme 5. Komplexe Suchanfragen 6. Sicherheitsmechanismen für Peer-to-Peer-Systeme 7. Software-Rahmenwerke für Peer-to-Peer-Systeme 		
14. Literatur:	<ul style="list-style-type: none"> • J.F. Kurose, K.W. Ross, Computer Networking. 4th Edition, 2007 • L.L. Peterson, B.S. Davie, Computer Networks: A Systems Approach. 4th Edition, 2007 • Peter Mahlmann, Christian Schindelbauer, P2P Netzwerke: Algorithmen und Methoden., 2007 		

Modul: 48580 Reinforcement Learning

2. Modulkürzel:	051200888	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	Vien Ngo		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Solid knowledge in linear algebra, probability theory and optimization. Rough knowledge of Artificial Intelligence. Fluency in at least one programming language.		
12. Lernziele:	Students will acquire a deep understanding of Reinforcement Learning methods. Reinforcement Learning addresses the problem of learning optimal behavior (strongly related to optimal control) from data. This course will enable students to apply Reinforcement Learning algorithms in simulated domains and real robotic systems.		
13. Inhalt:	<p>Reinforcement Learning considers how an agent, interacting with a world, can improve or learn optimal behavior based on own experience or teacher demonstration. This branch of Artificial Intelligence and Machine Learning has become increasingly important as a foundation of robust intelligent systems and robotics. Optimal exploration (behavior that optimizes the agent's information gain) is a particularly interesting aspect of Reinforcement Learning. This lecture will introduce to the theory of Reinforcement Learning and then discuss state-of-the-art algorithms in this area.</p> <p>motivation and history Markov Decision Processes and Bellman's optimality principle relations to stochastic optimal control theory basic model-free RL methods (TD-Learning, Q-learning, etc) model-based RL methods theory of optimal exploration (Bayesian RL, R-max) relational RL inverse RL, learning from demonstration and instruction information theoretic formulations of RL modern policy search methods (and applications in robotics)</p>		
14. Literatur:	<ul style="list-style-type: none"> - (Main background) R. Sutton and A. Barto, Reinforcement Learning, 1998. This book is freely available online. - (For robotics application) S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, 2006. - (Hardcore theory) C. Szepesvari, Algorithms for Reinforcement Learning, 2010. Draft version is freely available online. - S. LaValle, Planning Algorithms, 2006. http://planning.cs.uiuc.edu/ 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 485801 Lecture Reinforcement Learning • 485802 Exercise Reinforcement Learning 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		

17. Prüfungsnummer/n und -name: 48581 Reinforcement Learning (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Parallele und Verteilte Systeme

Modul: 48600 Robotics I

2. Modulkürzel:	051200999	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	Marc Toussaint		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Solid knowledge in linear algebra, probability theory and optimization. Fluency in at least one programming language.		
12. Lernziele:	Students will acquire the basic methodologies to model, control and navigate robots, including trajectory planning, control of dynamic systems and object manipulation.		
13. Inhalt:	<p>The lecture will give an introduction to robotics, focusing on essential theoretical foundations of planning and controlling motion, state estimation and eventually object manipulation. Exercises in simulations and on a real robot are a core element of this lecture to gain practical experience.</p> <ul style="list-style-type: none"> • motivation and history • (inverse) kinematics • path finding and trajectory optimization • (non-)holonomic systems • mobile robots • sensor processing (vision, range sensors) • simulation of robots and environments • object grasping and manipulation 		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 486001 Lecture Robotics I • 486002 Exercise Robotics I 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	48601 Robotics I (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Parallele und Verteilte Systeme		

Modul: 48620 Scientific Visualization

2. Modulkürzel:	051900777	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Filip Sadlo • Daniel Weiskopf 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2009 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Katalog ISW</p> <p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	<p>Basic concepts of Human Computer Interaction</p> <p>Basic concepts of Computer Graphics</p>		
12. Lernziele:	<p>Student gains expertise about fundamental concepts and techniques of scientific visualization. This includes algorithms and mathematical background, data structures and implementation aspects as well as practical experience with widely available visualization tools.</p>		
13. Inhalt:	<p>Visualization discusses all aspects of visual representations of data gained from experiments, simulations, medical scanning machines, data bases an the like. The aim of visualization is to gain further insights into the data or the generate "simple" representations of complex phenomena or issues. For that, known techniques from the research area of interactive computer graphics as well as novel techniques are applied.</p> <p>The following topics will be discussed:</p> <p>Introduction, history, visualization pipeline Data acquisition and representation (sampling, reconstruction, grids, data structures) Perception Basic concepts of visual mappings Visualization of scalar fields (extraction of iso-surfaces, volume rendering) Visualization of vector fields (particle tracking, texture-based methods, topology) Tensor fields, multivariate data Highdimensional data and information visualization</p>		
14. Literatur:	<p>C. D. Hansen, C. R. Johnson, The Visualization Handbook, 2005 C. Ware, Information Visualization: Perception for Design, 2004</p>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 486201 Lecture Scientific Visualization • 486202 Exercise Scientific Visualization 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden		

Selbststudium: 138 Stunden

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

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 29510 Service Computing

2. Modulkürzel:	052010004	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> • Frank Leymann • Dimka Karastoyanova 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	<p>A diversity of technologies enable nowadays computer-based interactions on the Web and on the Internet. The aim of this course is to make the students familiar with some of the most pervasive technologies that come together to form the Web and the Internet as we know it, and that enable to build large-scale application systems.</p>		
13. Inhalt:	<p>At first, we will cover the Web-centric technologies that enable the interaction of humans with Web content, e.g. HTTP, SMTP, AJAX, CSS and MIME . On the server-side part of technology, we will treat several Java EE technologies such as portlets, servlets, and JSP.</p> <p>The second part of the course will cover a set of technologies that are prominent in the landscape of Service-Oriented Architecture (SOA). In a nutshell, SOA is a paradigm that advocates the creation of complex, value added applications by reusing and composing independent and loosely coupled (software) services. We will dissect prominent SOA concepts like service discovery, addressing, policies, Service Bus, coordination protocols and service compositions. The architectural concepts will be complemented with an outlook of the technologies that embody them in the landscape of enterprise computing. In particular, we will cover several XML-centric technologies that sit at the core of Web services, e.g. XSD, SOAP, WSDL and Policy. In addition to the SOAP-based approach to Web services, we will also explore their REST aspect. Building on this portfolio of technologies, we will discuss the relationships between Web service technologies and “hot” items on the enterprise computing agenda such as autonomic/organic computing and cloud computing.</p>		
14. Literatur:	<p>S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson: “Web Services Platform Architecture”, Prentice Hall 2005</p> <p>G. Alonso, F. Casati, H. Kuno, V. Machiraju: “Web Services”, Springer 2004</p> <p>E. Wilde: “World Wide Web”, Springer 1999</p> <p>M.P. Papazoglou: “Web Services: Principles & Technology”, Pearson Education Limited 2008</p> <p>N.M. Josuttis: “SOA in Practice: The Art of Distributed System Design”, O’Reilly 2007</p>		

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

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

15. Lehrveranstaltungen und -formen:	295101 Vorlesung mit Übungen Service Computing
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 29511 Service Computing (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0• V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	29520 Ausgewählte Themen des Service Computing
19. Medienform:	Lecture and accompanying exercises
20. Angeboten von:	Architektur von Anwendungssystemen

Modul: 46660 Service Management and Cloud Computing, and Evaluation

2. Modulkürzel:	052000111	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Frank Leymann		
9. Dozenten:	Kristof Klöckner		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Service Computing, Business Process Management		
12. Lernziele:	The students will learn the basics of systems management and cloud computing.		
13. Inhalt:	<p>Cloud Computing is an emerging paradigm for consumption and delivery of IT based services, based on concepts derived from consumer internet services, like self-service, apparently unlimited or elastic resources and flexible sourcing options. In this course we will discuss the technical foundations of cloud computing, as well as the business models associated with it.</p> <p>We will start by looking at virtualization and service management as the technical underpinnings. We will then look at infrastructure services and platform services, with a particular focus on emerging programming models for the cloud. We will discuss the trade-offs made between consistency and availability as well as extensions to "traditional" programming models. We also look at the life-cycle of applications in the cloud.</p> <p>Finally, we will look some of the challenges of Software as a Service, like multi-tenancy.</p> <p>Throughout the course, we will look both at existing products and services as well as the theoretical underpinnings.</p> <p>The course will be held as a combination of lectures and participant discussion.</p>		
14. Literatur:	To be announced in the lecture.		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 466601 Vorlesung Service Management and Cloud Computing, and Evaluation • 466602 Excercise Service Management and Cloud Computing, and Evaluation 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 46661 Service Management and Cloud Computing, and Evaluation (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min. 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Architektur von Anwendungssystemen		

Modul: 42520 Services and Service Composition

2. Modulkürzel:	052010008	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Dimka Karastoyanova		
9. Dozenten:	<ul style="list-style-type: none"> • Vasilios Andrikopoulos • Dimka Karastoyanova • Frank Leymann 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	<p>The students will learn the foundations of the SOA and REST Architectural styles and technologies that can be used for their realization. The concept of service and the principle of loose coupling will be clarified. The students will be able to realize Service based applications using the Web Service technology. The students will be knowledgeable of the concepts workflow, service composition and how to apply them using workflow languages in order to create complex, value-added applications.</p>		
13. Inhalt:	<p>Architectural styles: SOA and REST Basic principles: loose coupling vs. tight coupling Service Technologies (WSDL, Policy, WS-Addressing, SOAP) Virtualization and Middleware (Service Bus, etc.) Basics of the Workflow Technology Business Process Re-engineering Workflow Life Cycle Workflow Management System Architecture Workflow Languages (FDL, BPEL)</p>		
14. Literatur:	<ul style="list-style-type: none"> • F. Leymann, D. Roller, Production Workflow, 2000 • S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, Web Services Platform Architecture, 2005 • W. van der Aalst, K. van Hee, Workflow Management, 2002 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 425201 Vorlesung Services and Service Compositions • 425202 Übung Services and Service Compositions 		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 48 Stunden Selbststudiumszeit: 132 Stunden</p>		
17. Prüfungsnummer/n und -name:	42521 Services and Service Composition (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Architektur von Anwendungssystemen		

Modul: 56550 Software Verification

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof. Lars Grunske		
9. Dozenten:	Antonio Filieri		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Theoretical CS and Software Engineering modules (eg. SEfST, SZS)		
12. Lernziele:	<p>The course has the objective to provide theoretical and practical knowledge about the essential software verification techniques. The goal is to enable students to use and apply them successfully to real world code artifacts. In this respects students will understand the required concepts of the different software verification techniques. Of great importance are, beside standard and time verification techniques also probabilistic verification techniques - these will be explained in detail and students will be able to apply them for quality assurance of probabilistic properties like safety, reliability and performance.</p>		
13. Inhalt:	<p>This course will introduce the foundations of this area</p> <p>Process algebraic specification software systems</p> <p>Abstract interpretation and symbolic execution</p> <p>Axiomatic semantics of software systems</p> <p>Software model checking</p> <p>Modeling time in computation and timed verification for software systems</p> <p>Quantitative analysis and probabilistic model checking</p> <p>Probabilistic symbolic execution</p> <p>Statistical methods in software verification</p>		
14. Literatur:	<p>1) Michael Huth and Mark Ryan. Logic in Computer Science: Modelling and Reasoning about Systems, second edition. Cambridge University Press, 2004.</p> <p>2) Christel Baier and Joost-Pieter Katoen. Principles of Model Checking. The MIT Press, 2008</p>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 565501 Vorlesung Software Verification • 565502 Übung Software Verification 		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden</p> <p>Nachbearbeitungszeit: 138 Stunden</p>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 56551 Software Verification (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0 		

-
- V Vorleistung (USL-V), schriftlich, eventuell mündlich
-

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 48640 Theoretical and Methodological Foundations of Autonomous Systems

2. Modulkürzel:	051200987	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Marc Toussaint		
9. Dozenten:	Marc Toussaint		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Solid knowledge in linear algebra, probability theory and optimization. Fluency in at least one programming language.		
12. Lernziele:	Students will acquire a conceptual overview of the challenges and research in intelligent autonomous systems. The course will emphasize the necessity of combining theory with integrated systems, namely the theoretical and computational foundations modeling and solving decision and behavioral problems and the integration in real-world autonomous systems that integrate perception, action and (on-board) computation. The course reflects the conceptual structure of the Major in Autonomous Systems by addressing the methodological foundations of (i) Computational Intelligence and Learning, (ii) Perception and Action, and (iii) System Integration.		
13. Inhalt:	This course discusses the challenges and research in intelligent autonomous systems. It introduces to the basic foundations in the relevant disciplines to enable a holistic view on autonomous systems. This is done using a coherent formalization for concepts which are usually introduced separately. <ul style="list-style-type: none"> • motivation and history • challenges in autonomous systems • frameworks for modeling decision and behavioral problems • computational methods for solving such problems: planning, decision making • system integration • classical Artificial Intelligence and modern probabilistic AI • perception and image processing • learning from data (basic regression and classification) • learning applied in autonomous systems (Reinforcement Learning, adaptive control, system identification) 		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 486401 Lecture Theoretical and Methodological Foundations of Autonomous Systems • 486402 Exercise Theoretical and Methodological Foundations of Autonomous Systems 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		

17. Prüfungsnummer/n und -name: 48641 Theoretical and Methodological Foundations of Autonomous Systems (PL), schriftlich oder mündlich, 120 Min.,
Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: Institut für Parallele und Verteilte Systeme

Modul: 46760 Theoretical and Methodological Foundations of Visual Computing

2. Modulkürzel:	051900022	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> • Thomas Ertl • Daniel Weiskopf • Thomas Müller • Andrés Bruhn 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT</p> <p>B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module</p> <p>M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	<p>Module der Mathematik, Numerik und Stochastik aus dem BSc Informatik oder BSc Softwaretechnik:</p> <ul style="list-style-type: none"> • 10190 Mathematik für Informatiker und Softwaretechniker • 10240 Numerische und Stochastische Grundlagen <i>oder</i> • 41590 Einführung in die Numerik und Stochastik für Softwaretechniker 		
12. Lernziele:	<p>Die Studierenden kennen die mathematisch-theoretischen Grundlagen des Visual Computing und können diese in Form von Methoden für die Computergraphik, Visualisierung, Bildverarbeitung und Computer Vision praktisch umsetzen.</p>		
13. Inhalt:	<p>Die Vorlesung behandelt die Grundlagen der affinen und projektiven Geometrie und deren Umsetzung in der Computergraphik, insbesondere innerhalb der Grafikpipeline. Es wird die Differential- und Integralrechnung und deren Anwendung in zwei und drei Dimensionen behandelt. Grundlagen der Theorie der gewöhnlichen und partiellen Differentialgleichungen werden vermittelt. Interpolations- und Approximationsverfahren werden im Kontext von Visual Computing vertieft. Methoden der Fourier-Analyse sowie der diskreten Wavelet-Analyse und deren Anwendung in der Bildverarbeitung werden behandelt. Übungen vertiefen den theoretischen Vorlesungsstoff und dienen auch als praktische Einführung in die Umsetzung der Methoden für numerische Berechnungen und Algorithmen der Computergraphik, Visualisierung, Bildverarbeitung und Computer Vision.</p> <p>Im Einzelnen werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none"> • Affine und projektive Geometrie: affiner Raum, affine Abbildung, orthographische und perspektivische Projektion, projektiver Raum, projektive Abbildung, homogene Koordinaten, Umsetzung in der Grafikpipeline • Differential- und Integralrechnung: partielle Ableitung, Gradient, Extrema in mehreren Variablen, numerische Ableitung, 		

- Kantendetektion, Taylor-Entwicklung in mehreren Variablen, vektorwertige Funktionen, Integralrechnung in mehreren Variablen
- Gewöhnliche und partielle Differentialgleichungen: Existenz und Eindeutigkeit, autonome Systeme, Vektorfelder, Integralkurven, numerische Verfahren
 - Interpolation und Approximation: Lagrange-Interpolation, Interpolation höherer Ordnung, baryzentrische Koordinaten, radiale Basisfunktionen, Shepard, Moving Least Squares (MLS), Kriging
 - Fourier-Analysis: kontinuierliche und diskrete Fourier-Transformation, Frequenz- und Phasenspektrum, Gibbs, Faltung, Dirac-delta, Abtasttheorem, diskrete Filter, Anwendungen in der Bildverarbeitung
 - Wavelet-Transformation: Haar-Transformation und -Wavelet, Multiresolution-Analyse, Daubechies-Wavelets, Denoising, Bildverarbeitung
 - Einführung in ein Softwaresystem zur praktischen Umsetzung (z.B. Matlab)

14. Literatur:	<ul style="list-style-type: none"> • B. Jähne. Digitale Bildverarbeitung. Springer, 2005 • H. Fischer, H. Kaul. Mathematik für Physiker - Band 1: Grundkurs. 5. Auflage, Teubner, 2005 • H. Fischer, H. Kaul. Mathematik für Physiker - Band 2: Gewöhnliche und partielle Differentialgleichungen, mathematische Grundlagen der Quantenmechanik. 2. Auflage, Teubner, 2004 • H. R. Schwarz, N. Köckler. Numerische Mathematik. 6. Auflage, Teubner, 2006 • J. S. Walker. A primer on WAVELETS and Their Scientific Applications. Chapman & Hall/CRC, 2008 • M. Oberguggenberger, A. Ostermann. Analysis für Informatiker. Springer, 2009 • J. Encarnação, W. Straßer, R. Klein. Graphische Datenverarbeitung 1. Oldenburg Verlag, 1996
15. Lehrveranstaltungen und -formen:	467601 Vorlesung Theoretische und Methodische Grundlagen des Visual Computing
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden, Selbststudium: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 46761 Theoretical and Methodological Foundations of Visual Computing (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0, schriftlich 120 Min. oder mündlich 30 Min. • V Vorleistung (USL-V), schriftlich, eventuell mündlich, Erfolgreiche Teilnahme an Übungen
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme

Modul: 29500 Visual Computing

2. Modulkürzel:	051900014	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Martin Fuchs		
9. Dozenten:	Martin Fuchs		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Ergänzungsmodule → Wahlmodule aus Master SWT B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	Modul 051900002 Computergraphik		
12. Lernziele:	The students know theoretical foundations for visual computing and acquired practical expertise in its core techniques. They are able to acquire scenes with digital cameras, can model their behavior and create content for non-2D displays and camera-projector systems.		
13. Inhalt:	The class is concerned with the digital processing of visual information by means of computer vision, computer graphics and image processing. It covers the following three interlocking topic complexes: Image processing: <ul style="list-style-type: none"> • mathematical basics of image representations • noise models and noise suppression (including morphological, bilateral, and non-local filters) • selected topics from discrete image processing on image regions (e.g. photo montage with graph cuts, texture synthesis and space-time video completion) Measuring / displaying light: <ul style="list-style-type: none"> • selected topics from simple optics (esp. thin lenses and their interactions with light) • geometric camera models and calibration, typical optical distortions and means to counter them • radiometric camera calibration and HDR imaging • measuring and displaying color • plenoptic imaging / integral photography techniques, light field rendering and light field displays • passive stereo Combined camera / illumination systems <ul style="list-style-type: none"> • camera - illumination systems and photometric stereo • active stereo and projector-camera systems • the light transport matrix, its measurement and applications Throughout, the class equally covers both acquisition (camera) and displays systems.		
14. Literatur:	<ul style="list-style-type: none"> • Andrew S. Glassner, Principles of Digital Image Synthesis, 1995 • J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics: Principle and Practice, 1990 • Jähne, Bernd, Digitale Bildverarbeitung, 2005 • Literatur, siehe Webseite zur Veranstaltung • M. Pharr, G. Humphreys, Physically Based Rendering, 2004 		

15. Lehrveranstaltungen und -formen:	295001 Vorlesung mit Übungen Visual Computing
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden
	Selbststudium: 138 Stunden
	Gesamt: 180 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 29501 Visual Computing (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündlichen 30 Min• V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

300 Fachaffine Schlüsselqualifikation

Zugeordnete Module: 42840 Software-Recht
 42850 Internetrecht

Modul: 42850 Internetrecht

2. Modulkürzel:	9500021	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Hon. Prof. Volker Haug		
9. Dozenten:	Volker Haug		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Fachaffine Schlüsselqualifikation		
11. Empfohlene Voraussetzungen:	keine		
12. Lernziele:	<ul style="list-style-type: none"> • Rechtsstruktur des Internet • Haftung von Internetanbietern (Caching, Hosting, Foren, Links, Anschlussinhaber etc.) • Internetstrafrecht (Viren, Hacking, Domain grabbing, ...) • Datenschutz und Globale Netzöffentlichkeit • Meine Domain, deine Domain - Namensrecht im Internet • Rechtsprobleme des eCommerce (Verbraucherschutz, Internetauktionen, Signaturregelung) • EGovernment (Internetwahlen, eCampaigning, elektronisches Rathaus etc.) 		
13. Inhalt:	Haug, Internetrecht, 2. Auflage, 2010		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	428501 Vorlesung Internetrecht		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 28 Stunden, Nachbearbeitungszeit: 56 Stunden		
17. Prüfungsnummer/n und -name:	42851 Internetrecht (USL), schriftlich, eventuell mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 42840 Software-Recht

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Hon. Prof. Volker Haug		
9. Dozenten:	Volker Haug		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Softwaretechnik, PO 2012 → Vorgezogene Master-Module M.Sc. Softwaretechnik, PO 2012 → Fachaffine Schlüsselqualifikation		
11. Empfohlene Voraussetzungen:	keine		
12. Lernziele:	Die Studierenden gewinnen einen Überblick über die wichtigsten Felder des Softwarerechts (s.u., Inhalt) und sind dadurch in der Lage, rechtliche Problemstellungen früher zu erkennen und ihnen durch geeignete Maßnahmen vorzubeugen. Zugleich können Sie bei auftretenden Rechtsfragen eine erste Einordnung vornehmen.		
13. Inhalt:	Nach einer terminologischen Klärung des rechtlichen Software-Begriffs werden in einem ersten Block die wichtigsten Schutzrechte für Software überblicksartig und mit besonderen Bezügen zu Softwarefragen vorgestellt, insbesondere der Urheber- und Patentrechtsschutz sowie der Markenrechtsschutz. Der zweite Hauptteil der Vorlesung befasst sich mit dem Software-Vertragsrecht, wobei es hier zunächst um verschiedene Vertragstypen mit spezifischen Problemstellungen geht (Kauf, Leasing, Miete, u.a.), bevor das Leistungsstörungenrecht zu den verschiedenen denkbaren Mängeln bei Softwareprodukten und ihrer Pflege behandelt wird.		
14. Literatur:	Wird in der Vorlesung bekannt gegeben		
15. Lehrveranstaltungen und -formen:	428401 Vorlesung Software-Recht		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 28 Stunden, Nachbearbeitungszeit: 56 Stunden, Gesamt 84 Stunden		
17. Prüfungsnummer/n und -name:	42841 Software-Recht (USL), schriftlich, eventuell mündlich, 30 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:	<ul style="list-style-type: none"> • Vortrag • PowerPoint-Folien • Tafelanschiebe 		
20. Angeboten von:			

Modul: 80620 Masterarbeit-SWT

2. Modulkürzel:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	30.0 LP	6. Turnus:	jedes Semester
4. SWS:	0.0	7. Sprache:	Englisch

8. Modulverantwortlicher:

9. Dozenten:

10. Zuordnung zum Curriculum in diesem Studiengang:

11. Empfohlene Voraussetzungen:

12. Lernziele:

13. Inhalt:

14. Literatur:

15. Lehrveranstaltungen und -formen:

16. Abschätzung Arbeitsaufwand:

17. Prüfungsnummer/n und -name:

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:
