



**Universität Stuttgart**

**Modulhandbuch**  
**Studiengang Master of Science Softwaretechnik**  
**Prüfungsordnung: 2012**

Wintersemester 2012/13  
Stand: 20. November 2012

Universität Stuttgart  
Keplerstr. 7  
70174 Stuttgart

## Kontaktpersonen:

---

Studiengangsmanager/in:

Corinna Vehlow  
Visualisierungsinstitut der Universität Stuttgart  
Tel.:  
E-Mail: [corinna.vehlow@visus.uni-stuttgart.de](mailto:corinna.vehlow@visus.uni-stuttgart.de)

## Inhaltsverzeichnis

<b>100 Vertiefungsmodule .....</b>	<b>5</b>
120 Vertiefungslinien .....	6
46450 Vertiefungslinie Architektur von Anwendungssystemen .....	7
29330 Vertiefungslinie Datenbanken und Informationssysteme .....	8
46470 Vertiefungslinie Parallele Systeme .....	10
46440 Vertiefungslinie Programmiersprachen, Compilerbau, Programmanalysen .....	11
29370 Vertiefungslinie Rechnerarchitekturen und eingebettete Systeme .....	12
46460 Vertiefungslinie Sprachverarbeitung .....	13
29380 Vertiefungslinie Theoretische Informatik und Wissenschaftliches Rechnen .....	14
29390 Vertiefungslinie Verteilte Systeme .....	15
29400 Vertiefungslinie Visualisierung und Interaktive Systeme .....	16
110 Vertiefungsmodule Pflicht .....	17
42830 Entwicklungsprojekt .....	18
33350 Forschungsmethoden in der Softwaretechnik .....	19
42860 Hauptseminar (Master SWT 1) .....	20
55560 Hauptseminar (Master SWT 2) .....	21
42820 Prozessanalyse .....	22
36410 Requirements Engineering und Software-Architektur .....	23
42810 Software-Qualitätssicherung und -Wartung .....	24
 <b>200 Spezialisierungsmodule - MSWT .....</b>	 <b>25</b>
55740 Advanced Service Computing .....	26
29550 Algorithmische Geometrie .....	27
29560 Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems .....	28
10040 Bildsynthese .....	30
29570 Computer Interface Technologien .....	32
29430 Computer Vision .....	33
29580 Datenkompression .....	35
29590 Digitale Systeme .....	36
29600 Digitale Systeme II .....	37
29710 Embedded Systems Engineering .....	38
29740 Fachpraktikum Eingebettete Systeme .....	39
29440 Geometrische Modellierung und Animation .....	40
29450 Graphentheorie .....	42
29610 Hardware Based Fault Tolerance .....	43
42920 Hardware-Software-Codesign .....	45
55630 Informationsvisualisierung .....	46
29630 Konzepte der Programmiersprachen .....	48
29460 Kryptographische Verfahren .....	50
42870 Message-Basierte Anwendungen .....	51
29640 Mikrocontroller .....	52
29720 Mobile Computing .....	54
29730 Modelling, Simulation, and Specification .....	56
29650 Parallele Programmierung .....	58
29660 Programmanalysen und Compilerbau .....	59
29670 Rapid Prototyping .....	61
29680 Real-Time Programming .....	62
29690 Real-Time Video Processing I .....	63
29700 Real-Time Video Processing II .....	64
29510 Service Computing .....	65
42520 Services und Service Komposition .....	67
46760 Theoretische und Methodische Grundlagen des Visual Computing .....	68
29500 Visual Computing .....	70
11330 Visualisierung .....	72

---

42900 Workflow Management 1 .....	74
42910 Workflow Management 2 .....	75
<b>300 Fachaffine Schlüsselqualifikation .....</b>	<b>76</b>
42850 Internetrecht .....	77
42840 Software-Recht .....	78
<b>80620 Masterarbeit-SWT .....</b>	<b>79</b>

---

## 100 Vertiefungsmodule

---

Zugeordnete Module:	110	Vertiefungsmodule Pflicht
	120	Vertiefungslinien

---

---

## 120 Vertiefungslinien

---

Zugeordnete Module:

- 29330 Vertiefungslinie Datenbanken und Informationssysteme
- 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:	-	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Frank Leymann		
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"> <li>• 464501 Deutschsprachige Lehrveranstaltungen Wintersemester</li> <li>• 464502 Deutschsprachige Lehrveranstaltungen Sommersemester</li> <li>• 464503 courses in english - winter semester</li> <li>• 464504 courses in english - summer semester</li> </ul>		
16. Abschätzung Arbeitsaufwand:			
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:			

## 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 Ankündigung
8. Modulverantwortlicher:	Prof.Dr.-Ing. Bernhard Mitschang		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Bernhard Mitschang</li> <li>• Holger Schwarz</li> </ul>		
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"> <li>1) Datenbanken und Informationssysteme (Vorlesung mit Übung, 4 SWS)</li> <li>2) Advanced Information Management (Vorlesung mit Übung, 4 SWS)</li> <li>3) Data Warehousing, Data Mining und OLAP-Technologien (Vorlesung mit Übung, 4 SWS)</li> <li>4) Information Integration (Vorlesung mit Übung, 4 SWS)</li> </ol>		
14. Literatur:	<ul style="list-style-type: none"> <li>• A. Kemper, A. Eickler, Datenbanksysteme - Eine Einführung, 2004</li> <li>• T. Härder, E. Rahm, Datenbanksysteme - Konzepte und Techniken der Implementierung, 2001</li> </ul> Weitere Literatur wird in den einzelnen Lehrveranstaltungen bekanntgegeben.		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 293301 Deutschsprachige Lehrveranstaltungen Wintersemester</li> <li>• 293302 Deutschsprachige Lehrveranstaltungen Sommersemester</li> <li>• 293303 courses in english - winter semester</li> <li>• 293304 courses in english - summer semester</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 84 Stunden Selbststudium: 276 Stunden  <b>Gesamt: 360 Stunden</b>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29331 Vertiefungslinie Datenbanken und Informationssysteme (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0</li> <li>• 29332 Vorleistung (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0</li> </ul>		
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:	Prof.Dr.-Ing. Sven Simon		
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"> <li>• 464701 Deutschsprachige Lehrveranstaltungen Wintersemester</li> <li>• 464702 Deutschsprachige Lehrveranstaltungen Sommersemester</li> <li>• 464703 courses in english - winter semester</li> <li>• 464704 courses in english - summer semester</li> </ul>		
16. Abschätzung Arbeitsaufwand:			
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 2. Semester, SoSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Erhard Plödereder		
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"> <li>• 464401 Deutschsprachige Lehrveranstaltungen Wintersemester</li> <li>• 464402 Deutschsprachige Lehrveranstaltungen Sommersemester</li> <li>• 464403 courses in english - winter semester</li> <li>• 464404 courses in english - summer semester</li> </ul>		
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:			

## 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:	Prof.Dr. Hans-Joachim Wunderlich		
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"> <li>• 293701 Deutschsprachige Lehrveranstaltungen Wintersemester</li> <li>• 293702 Deutschsprachige Lehrveranstaltungen Sommersemester</li> <li>• 293703 courses in english - winter semester</li> <li>• 293704 courses in english - summer semester</li> </ul>		
16. Abschätzung Arbeitsaufwand:			
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:			

## 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:	Prof.Ph.D. Hinrich Schütze		
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"> <li>• 464601 Deutschsprachige Lehrveranstaltungen Wintersemester</li> <li>• 464602 Deutschsprachige Lehrveranstaltungen Sommersemester</li> <li>• 464603 courses in english - winter semester</li> <li>• 464604 courses in english - summer semester</li> </ul>		
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 46461 Vertiefungslinie Sprachverarbeitung (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>		
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:	Deutsch
8. Modulverantwortlicher:		Prof.Dr. Volker Diekert	
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"> <li>• 293801 Deutschsprachige Lehrveranstaltungen Wintersemester</li> <li>• 293802 Deutschsprachige Lehrveranstaltungen Sommersemester</li> <li>• 293803 courses in english - winter semester</li> <li>• 293804 courses in english - summer semester</li> <li>• 293805 Vorlesung mit Übung Quantencomputing</li> </ul>	
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:		<ul style="list-style-type: none"> <li>• 29381 Vertiefungslinie Theoretische Informatik und Wissenschaftliches Rechnen (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0</li> <li>• 29382 Vorleistung (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0</li> </ul>	
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## 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:	Prof.Dr. Kurt Rothermel		
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"> <li>• 293901 Deutschsprachige Lehrveranstaltungen Wintersemester</li> <li>• 293902 Deutschsprachige Lehrveranstaltungen Sommersemester</li> <li>• 293903 courses in english - winter semester</li> <li>• 293904 courses in english - summer semester</li> <li>• 293905 Vorlesung Reliable Distributed Programming</li> <li>• 293906 Vorlesung Peer-to-Peer Systeme</li> <li>• 293907 Vorlesung Asynchrone Middleware-Systeme</li> <li>• 293908 Vorlesung Selbstorganisation in Verteilten Systemen</li> </ul>		
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29391 Vertiefungslinie Verteilte Systeme (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0</li> <li>• 29392 Vorleistung (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## 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:	Prof.Dr. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> <li>• Martin Fuchs</li> </ul>		
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"> <li>• Grundlegende Kenntnisse in Computergraphik und Bildverarbeitung (z.B. Computergraphik 051900002 und Imaging Science 051900210)</li> </ul>		
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: a) Bildsynthese (Vorlesung mit Übung, 4 SWS) b) Geometrische Modellierung und Animation (Vorlesung mit Übung, 4 SWS) c) Visual Computing (Vorlesung mit Übung, 4 SWS) d) Visualisierung (Vorlesung mit Übung, 4 SWS)		
14. Literatur:	Andrew S. Glassner, Principles of Digital Image Synthesis, 1995		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 294001 Deutschsprachige Lehrveranstaltungen Wintersemester</li> <li>• 294002 Deutschsprachige Lehrveranstaltungen Sommersemester</li> <li>• 294003 courses in english - winter semester</li> <li>• 294004 courses in english - summer semester</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 84 Stunden Selbststudium: 276 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29401 Vertiefungslinie Visualisierung und Interaktive Systeme (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0</li> <li>• 29402 Vorleistung (PL), mündliche Prüfung, 45 Min., Gewichtung: 1.0</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme		



---

## 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.Dr. 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:	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"> <li>• Erstellung einer detaillierten Spezifikation</li> <li>• Entwicklung eines fortgeschrittenen Softwaresystems</li> <li>• Änderung und Erweiterung eines bestehenden Systems</li> </ul>		
14. Literatur:			
15. Lehrveranstaltungen und -formen:	428301 Praktikum Entwicklungsprojekt		
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	42831 Entwicklungsprojekt (LBP), schriftliche Prüfung, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 33350 Forschungsmethoden in der Softwaretechnik

2. Modulkürzel:	[pord.modulcode]	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.Dr. 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"> <li>• Wissenschaftstheorie</li> <li>• Theoretische, methodische, konstruktive und empirische Forschung</li> <li>• Qualitative und quantitative Methoden</li> <li>• Systematische Literaturlauswertung, Umfragen, Interviews</li> <li>• Experimente und Fallstudien</li> <li>• Schreiben und publizieren</li> <li>• Deskriptive (beschreibende) Statistik</li> <li>• Inferentielle (induktive) Statistik</li> <li>• Wichtige Verteilungen und Hypothesentests</li> <li>• Fallbeispiele</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Shull, Singer, Sjøberg (Eds.). Guide to Advanced Empirical Software Engineering. Springer, 2008</li> <li>• Leedy, Ormrod. Practical Research: Planning and Design. Pearson Prentice Hall, 2009</li> <li>• Rosenkrantz. Introduction to Probability and Statistics for Scientists and Engineers. McGraw-Hill, 1997</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 333501 Vorlesung Forschungsmethoden in der Softwaretechnik</li> <li>• 333502 Übung Forschungsmethoden in der Softwaretechnik</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<ul style="list-style-type: none"> <li>• Vorlesung: Präsenzzeit: 31,5 Stunden, Nachbearbeitungszeit: 103,5</li> <li>• Übung: Präsenzzeit: 10,5 Stunden, Nachbearbeitungszeit: 34,5</li> </ul>		
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:			

## Modul: 42860 Hauptseminar (Master SWT 1)

2. Modulkürzel:	[pord.modulcode]	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes Semester
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof.Dr. Daniel Weiskopf		
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 Studierenden können wissenschaftliche Texte selbständig durcharbeiten, zusammenfassen und die wesentlichen Resultate präsentieren.		
13. Inhalt:	Es werden neuere wissenschaftliche Texte zu einem Gebiet der Informatik und Softwaretechnik in Vorträgen der Teilnehmer behandelt und im Plenum diskutiert.		
14. Literatur:	<ul style="list-style-type: none"> <li>• Deininger, Lichter, Ludewig, Schneider. Studien-Arbeiten, 5. Aufl., 2005</li> <li>• Franck, Norbert, Sary, Joachim. Die Technik wissenschaftlichen Arbeitens, 2007</li> </ul>		
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.Dr. Daniel Weiskopf		
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:	[pord.modulcode]	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes Semester
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof.Dr. 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"> <li>• Arbeit in Zweiertteams</li> <li>• Analyse und Dokumentation eines Entwicklungsprozesses einer Firma</li> <li>• Interviews und Umfragen</li> <li>• Verbesserungsvorschläge zum Prozess</li> <li>• Präsentation der Ergebnisse</li> </ul>		
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), schriftliche Prüfung, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 36410 Requirements Engineering und Software-Architektur

2. Modulkürzel:	[pord.modulcode]	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.Dr. 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"> <li>• Methoden des Requirements Engineerings</li> <li>• Beschreibung und Modellierung von Anforderungen</li> <li>• Analyse und Validierung von Anforderungen</li> <li>• Management von Anforderungen</li> <li>• Modellierung, Erstellung und Analyse von Software-Architekturen</li> <li>• Architekturmuster</li> <li>• Requirements Engineering und Architektur im Entwicklungsprozess</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Robertson, Robertson. Mastering the Requirements Process. Addison-Wesley Professional, 2006</li> <li>• Sommerville, Sawyer. Requirements Engineering: A Good Practice Guide. John Wiley &amp; Sons, 1997</li> <li>• Bass, Clements, Kazman. Software Architecture in Practice, 2nd edition. Addison-Wesley Professional, 2003</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 364101 Vorlesung Requirements Engineering und Software-Architektur</li> <li>• 364102 Übung Requirements Engineering und Software-Architektur</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<ul style="list-style-type: none"> <li>• Vorlesung Requirements Engineering und Software-Architektur: Präsenzzeit: 31,5 Stunden, Nachbearbeitungszeit: 103,5</li> <li>• Übung Requirements Engineering und Software-Architektur: Präsenzzeit: 10,5 Stunden, Nachbearbeitungszeit: 34,5</li> </ul>		
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:			

## 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.Dr. 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		
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 Anwendungskenntnisse.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Qualitätssicherungs- und Wartungsprozesse</li> <li>• Prozesskontrolle und -steuerung</li> <li>• Qualitätssicherungsmethoden und Qualitätsmodelle</li> <li>• Vorhersagemodelle</li> <li>• Programmanalyse und Programmverstehen</li> <li>• Werkzeugunterstützung</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Liggesmeyer. Software-Qualität. Testen, Analysieren und Verifizieren von Software. Spektrum Akademischer Verlag, 2002</li> <li>• Sneed, Hasitschka, Teichmann. Software-Produktmanagement. Wartung und Weiterentwicklung bestehender Anwendungssysteme. Dpunkt, 2004</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 428101 Vorlesung Software-Qualitätssicherung und -Wartung</li> <li>• 428102 Übung Software-Qualitätssicherung und -Wartung</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<ul style="list-style-type: none"> <li>• Vorlesung: Präsenzzeit: 31,5 Stunden, Nachbearbeitungszeit: 103,5</li> <li>• Übung: Präsenzzeit: 10,5 Stunden, Nachbearbeitungszeit: 34,5</li> </ul>		
17. Prüfungsnummer/n und -name:	42811 Software-Qualitätssicherung und -Wartung (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			



---

## 200 Spezialisierungsmodule - MSWT

---

Zugeordnete Module:	10040	Bildsynthese
	11330	Visualisierung
	29430	Computer Vision
	29440	Geometrische Modellierung und Animation
	29450	Graphentheorie
	29460	Kryptographische Verfahren
	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	Datenkompression
	29590	Digitale Systeme
	29600	Digitale Systeme 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
	42520	Services und Service Komposition
	42870	Message-Basierte Anwendungen
	42900	Workflow Management 1
	42910	Workflow Management 2
	42920	Hardware-Software-Codesign
	46760	Theoretische und Methodische Grundlagen des Visual Computing
	55630	Informationsvisualisierung
	55740	Advanced Service Computing

---

## Modul: 55740 Advanced Service Computing

2. Modulkürzel:	052010005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Frank Leymann		
9. Dozenten:			
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:	428901 Vorlesung mit Übungen, Web Services 2		
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 55741 Advanced Service Computing (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## 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:	Prof.Dr. 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:			

## 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:	Prof.Dr. 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"> <li>• G. De Micheli: Synthesis and Optimization of Digital Circuits, McGrawHill, New York, NY, USA, 1994.</li> <li>• Hachtel, G. D. and Somenzi: Logic Synthesis and Verification Algorithms, Kluwer Academic Publishers, Norwell, MA, USA, 2000.</li> <li>• Ban Wong, Anurag Mittal, Yu Cao: Nano-CMOS Circuit and Physical Design, John Wiley &amp; Sons Inc, 2004.</li> <li>• Ashish Srivastava, Dennis Sylvester, David Blaauw: Statistical Analysis and Optimization for VLSI: Timing and Power, Springer, 2005.</li> <li>• Jens Lienig: Layoutsynthese elektronischer Schaltungen - Grundlegende Algorithmen fuer die Entwurfsautomatisierung, Springer, 2006.</li> <li>• L.-T. Wang, Y.-W. Chang, K.-W. Cheng: Electronic Design Automation, Morgan Kaufmann, 2009</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 295601 Vorlesung Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems</li> <li>• 295602 Übung Algorithms and Methods in Design Automation for Micro- and Nanoelectronic Systems</li> </ul>		
16. Abschätzung Arbeitsaufwand:	<b>Präsenzzeit:</b>	<b>42 Stunden</b>	
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: 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:	Prof.Dr. Thomas Ertl		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Martin Fuchs</li> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> </ul>		
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>M.Sc. Softwaretechnik, PO 2012          → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• Modul 051900002 Computergraphik</li> </ul>		
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"> <li>• Grafik Hardware und APIs, OpenGL</li> <li>• Texturen, prozedurale Modelle</li> <li>• Schattenberechnungen</li> <li>• Szenengraphen, Culling, Level-of-Detail Verfahren</li> <li>• Physikalisch-basierte Beleuchtungsberechnung, Fotorealistische Bildsynthese</li> <li>• Lokale Beleuchtungsmodelle</li> <li>• Raytracing, Monte-Carlo Methoden</li> <li>• Radiosity</li> <li>• Bild-basiertes Rendering</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Andrew S. Glassner, Principles of Digital Image Synthesis, 1995</li> <li>• D. Eberly, 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, 2000</li> <li>• J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics: Principle and Practice, 1990</li> <li>• Literatur, siehe Webseite zur Veranstaltung</li> <li>• P. Dutre, P. Bekaert, K. Bala, Advanced Global Illumination, 2003</li> <li>• Tomas Akenine-Möller, Eric Haines, Real-Time Rendering, 2002</li> </ul>		

---

	<ul style="list-style-type: none"><li>• Matt Pharr, Greg Humphreys, Physically Based Rendering: From Theory To Implementation, Morgan Kaufmann; Auflage: 2nd revised edition. (26. August 2010)</li><li>• Peter Shirley et al, Fundamentals of Computer Graphics, Third Edition, A.K. Peters, July 2009</li></ul>
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 100401 Vorlesung Bildsynthese</li><li>• 100402 Übung Bildsynthese</li></ul>
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 10041 Bildsynthese (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich, Übungsschein.</li></ul>
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	Institut für Visualisierung und Interaktive Systeme

---

## 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 Ankündigung
8. Modulverantwortlicher:	Prof.Dr.-Ing. 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"> <li>• Grundlagen - Computer Interfaces</li> <li>• Computer Interfaces und OSI-Modelle</li> <li>• Bus- und Netz-Topologien</li> <li>• Line und Error Codes</li> <li>• Protokolle</li> <li>• Treiber</li> <li>• Compliance Tests</li> <li>• Standardization Groups: USB, PCI, etc.</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Patterson, David A.; Hennessey, John L., Computer Organization and Design - The Hardware / Software Interface, 2008</li> </ul> <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 <b>Gesamt: 180 Stunden</b>		
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.Dr.-Ing. Andrés Bruhn		
9. Dozenten:	Andrés Bruhn		
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:	<ul style="list-style-type: none"> <li>• Modul 080300100 Mathematik für Informatiker und Softwaretechniker</li> <li>• Modul 050700005 Imaging Science</li> </ul>		
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"> <li>• Lineare Diffusion, Skalenräume</li> <li>• Bildpyramiden, Kanten und Eckendetektion</li> <li>• Hough-Transformation, Invarianten</li> <li>• Texturanalyse</li> <li>• Scale Invariant Feature Transform (SIFT)</li> <li>• Bildfolgenanalyse: lokale Verfahren</li> <li>• Bewegungsmodelle, Objektverfolgung, Feature Matching</li> <li>• Bildfolgenanalyse: globale Verfahren</li> <li>• Kamerageometrie, Epipolargeometrie</li> <li>• Stereo Matching und 3-D Rekonstruktion</li> <li>• Shape-from-Shading</li> <li>• Isotrope und anisotrope nichtlineare Diffusion</li> <li>• Segmentierung mit globalen Verfahren</li> <li>• Kontinuierliche Morphologie, Schockfilter</li> <li>• Mean Curvature Motion</li> <li>• Self-Snakes, Aktive Konturen</li> <li>• Bayes'sche Entscheidungstheorie der Mustererkennung</li> <li>• Klassifikation mit parametrischen Verfahren, Dichteschätzung</li> <li>• Klassifikation mit nicht-parametrischen Verfahren</li> <li>• Dimensionsreduktion</li> </ul> <ul style="list-style-type: none"> <li>• Linear Diffusion, Scale Space</li> <li>• Image Pyramids, Edges and Corners</li> <li>• Hough Transform, Invariants</li> <li>• Texture Analysis</li> <li>• Scale Invariant Feature Transform</li> <li>• Image Sequence Analysis: Local Methods</li> <li>• Motion Models, Tracking, Feature Matching</li> <li>• Image Sequence Analysis: Variational Methods</li> </ul>		

- Camera Geometry, Epipolar Geometry
- Stereo Matching and 3-D Reconstruction
- Shape-from-Shading
- Isotropic and Anisotropic Nonlinear Diffusion
- Segmentation with Global Methods
- Continuous Scaled Morphology, Shock Filters
- Mean Curvature Motion
- Self-Snakes, Active Contours
- Bayes Decision Theory for Pattern Recognition
- Classification with Parametric Techniques, Density Estimation
- Classification with Non-Parametric Techniques
- Dimensionality Reduction

14. Literatur:

- Forsyth, David and Ponce, Jean, Computer Vision. A Modern Approach.: 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: 294301 Vorlesung mit Übungen Computer Vision

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

**Gesamt: 180 Stunden**

17. Prüfungsnummer/n und -name:

- 29431 Computer Vision (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein, Kriterien werden in der ersten Vorlesung bekannt gegeben
- V Vorleistung (USL-V), schriftlich, eventuell mündlich

18. Grundlage für ... : 55640 Correspondence Problems in Computer Vision

19. Medienform:

20. Angeboten von: Institut für Visualisierung und Interaktive Systeme

## Modul: 29580 Datenkompression

2. Modulkürzel:	051230110	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Nach Ankündigung
8. Modulverantwortlicher:	Prof.Dr.-Ing. 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:	Grundlagenkenntnisse im Fach Mathematik.		
12. Lernziele:	Die Studierenden haben die Konzepte der Datenkompression erlernt und haben Verständnis über verschiedene Algorithmen zur Datenkompression erworben. Sie können die behandelten Algorithmen implementieren und weiterentwickeln.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Shannons Entropie-Definition</li> <li>• Huffman Codierung</li> <li>• Arithmetische Codierung</li> <li>• Universal Codes</li> <li>• Verlustlose u. verlustbehaftete Kompression</li> <li>• Bilddatenkompression</li> <li>• Wörterbuch-basierte Kompression</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Khalid Sayood, Introduction to Data Compression, 2005</li> <li>• More literature is named in the lecture</li> </ul>		
15. Lehrveranstaltungen und -formen:	295801 Vorlesung mit Übung Datenkompression		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden <b>Gesamt: 180 Stunden</b>		
17. Prüfungsnummer/n und -name:	29581 Datenkompression (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündliche Prüfung von 30 Min.		
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 Ankündigung
8. Modulverantwortlicher:	Prof.Dr.-Ing. 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"> <li>• Praktische Einführung in den System-Entwurf mit digitalen Komponenten wie Schnittstellenbausteinen zur Kommunikation, FPGAs, Prozessoren, intelligenten Sensoren etc.</li> <li>• Einführung und Verwendung der Hardware-Beschreibungssprache VHDL zum Entwurf Digitaler Systeme</li> <li>• Digitale Systeme und Board-Integration von digitalen Komponenten</li> <li>• Aufbau von Computer-Boards u. Gbit/s-Interconnects</li> <li>• Entwurf auf höheren Abstraktionsebenen zur schnellen Entwicklung von Prototypen</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Kou-Chuan Chang, K.C. Chang, Digital Systems Design with VHDL and Synthesis: An Integrated Approach, 1999</li> </ul> <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 <b>Gesamt: 180 Stunden</b>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29591 Digitale Systeme (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündliche Prüfung von 30 Min.</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich, 120 Min.</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 29600 Digitale Systeme 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:	Prof.Dr.-Ing. 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 Digital System Design II sind Kenntnisse von Digital System Design I. Alternativ sind Kenntnisse aus einem Fach der Technischen Informatik oder einem ähnlichen Fach Voraussetzung.		
12. Lernziele:	Die Studierende können ein komplexes digitales System aufbauen und implementieren unter Verwendung von digitalen Komponenten auf einem Board und sie haben das vertiefte Wissen zur Realisierung komplexer digitaler Systeme mittels FPGAs erworben.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Vorstellung der Fallstudie eines digitalen Systems</li> <li>• Simulierbare Spezifikation des Systems</li> <li>• Architektur für die Realisierung mittels FPGAs</li> <li>• Entwurf und Entwurfswerkzeuge zur Board-Integration</li> <li>• Implementierung des digitalen Systems</li> <li>• Verifikation des digitalen Systems</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Kou-Chuan Chang, K. C. Chang, Digital Systems Design with VHDL and Synthesis: An Integrated Approach, 1999</li> <li>• More literature is named in the lecture</li> </ul>		
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, 120 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 29710 Embedded Systems Engineering

2. Modulkürzel:	051711027	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Prof.Dr.-Ing. 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:	Understanding of the design methodology and advanced design techniques for constructing and analyzing embedded hardware / software systems. Practical experience in utilizing and programming an embedded platform.		
13. Inhalt:	<ol style="list-style-type: none"> <li>1. Introduction to embedded systems and their design constraints</li> <li>2. High level synthesis, scheduling, allocation, binding</li> <li>3. Pipelined data path and controller design</li> <li>4. Software task scheduling and schedulability analysis</li> <li>5. Static and dynamic methods for scheduling and priority assignment</li> <li>6. Implementation architectures for embedded systems</li> <li>7. Communication architectures; bus and memory systems</li> <li>8. System synthesis; partitioning of specifications into hardware and software parts</li> <li>9. Integrated hands-on exercises covering microcontroller programming, hardware / software interaction and cyclic executive scheduling of software tasks</li> </ol>		
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"> <li>• 297101 Vorlesung Embedded Systems Engineering</li> <li>• 297102 Übung Embedded Systems Engineering</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden  <b>Summe: 180 Stunden</b>		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 29711 Embedded Systems Engineering (PL), schriftlich, eventuell mündlich, 120 Min., Gewichtung: 1.0</li> <li>• 29712 Embedded Systems Engineering (BSL), schriftliche Prüfung, Gewichtung: 1.0</li> </ul>		
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:	Prof.Dr.-Ing. Martin Radetzki		
9. Dozenten:			
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. 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:			
17. Prüfungsnummer/n und -name:	29741 Fachpraktikum Eingebettete Systeme (LBP), schriftlich oder mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 29440 Geometrische Modellierung und Animation

2. Modulkürzel:	051900010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Univ.-Prof.Dr. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Daniel Weiskopf</li> <li>• Thomas Ertl</li> </ul>		
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:	<ul style="list-style-type: none"> <li>• Modul 051900002 Computergraphik</li> <li>• Modul 051900001 Mensch-Computer-Interaktion</li> <li>• Modul 051240005 Numerik und Stochastik</li> </ul>		
12. Lernziele:	Die Studierende kennen die theoretischen Grundlagen der geometrischen Modellierung und Animation und haben praktische Fähigkeiten in der Programmierung von Verfahren der geometrischen Modellierung und Animation sowie in der Benutzung von Modellierungs- und Animationssoftware wie Maya erlangt.		
13. Inhalt:	<p>Die Vorlesung behandelt Grundlagen und ausgewählte Verfahren der Szenenmodellierung und der Computeranimation. Dazu gehören Kenntnisse über Kurven- und Flächenrepräsentationen, wie sie in Animationspaketen zur Modellierung von Objekten, zur Beschreibung von dynamischen Verhalten von Parametern und zur Keyframe-Interpolation verwendet werden.</p> <p>Physikalisch-basierte Animation hingegen beschreibt Bewegung durch die kinematischen und dynamischen Gesetze der Mechanik. Anwendungen reichen von Partikelsystemen bis zur Simulation von mehrgliedrigen Modellen und Verformungen.</p> <p>Folgende Themen werde in der Vorlesung behandelt:</p> <ul style="list-style-type: none"> <li>• Kurvenbeschreibung und -modellierung (allgemeine Polynomkurven, Bezier-Kurven, B-Splines, NURBS)</li> <li>• Flächenmodellierung (Tensorproduktflächen, NURBS)</li> <li>• Unterteilungsschemata</li> <li>• Überblick über Animationstechniken</li> <li>• Keyframe-Animation</li> <li>• Physikalisch basierte Animation</li> </ul> <p>Die Veranstaltung besteht aus Vorlesung mit Übungen. Die Übungen umfassen praktische Programmierübungen und Anwendungen von Modellierungs- und Animationswerkzeugen (wie Maya) sowie theoretische Themen.</p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• D. Eberly, 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, 2000</li> <li>• G. Farin, Curves and Surfaces for CAGD: A Practical Guide, 2002</li> <li>• R. Parent, Computer Animation: Algorithms and Techniques, 2002</li> </ul>		



---

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"><li>• 29441 Geometrische Modellierung und Animation (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündlichen Prüfung von 30 Min. Written exam of 120 min. or oral exam of 30 min.</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

---

## Modul: 29450 Graphentheorie

2. Modulkürzel:	050420105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	unregelmäßig
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Prof.Dr. Volker Diekert		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Ulrich Hertrampf</li> <li>• Volker Diekert</li> <li>• Manfred Kufleitner</li> </ul>		
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"> <li>• Eulergraphen</li> <li>• Cographen</li> <li>• Bipartite Graphen</li> <li>• Planare Graphen, Eulerformel, Satz von Kuratowski</li> <li>• Graphparameter</li> <li>• Perfekte Graphen</li> <li>• Graphenfärbungen und der Satz von Ramsey</li> <li>• Extremale Graphentheorie</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Reinhard Diestel: Graphentheorie. Springer, 2010.</li> <li>• Martin Aigner, Günter M. Ziegler: Das BUCH der Beweise. Springer, 2009.</li> <li>• Jacobus H. van Lint, Richard M. Wilson: A Course in Combinatorics. Cambridge University Press, 2nd edition, 2001.</li> </ul>		
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"> <li>• 29451 Graphentheorie (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## 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:	Prof.Dr. Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Hans-Joachim Wunderlich</li> <li>• Michael Kochte</li> </ul>		
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"> <li>• 296101 Vorlesung Hardware Based Fault Tolerance</li> <li>• 296102 Übung Hardware Based Fault Tolerance</li> </ul>		
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, 60 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:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr.-Ing. 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"> <li>• 429201 Vorlesung Hardware-Software-Codesign</li> <li>• 429202 Übung Hardware-Software-Codesign</li> </ul>		
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:			

## Modul: 55630 Informationsvisualisierung

2. Modulkürzel:	051900099	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Nach Ankündigung
8. Modulverantwortlicher:	Univ.-Prof.Dr. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Michael Burch</li> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> </ul>		
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:	<ul style="list-style-type: none"> <li>• Mensch-Computer-Interaktion</li> <li>• Computergraphik</li> </ul>		
12. Lernziele:	Die Studierende kennen die theoretischen Grundlagen der Informationsvisualisierung und haben praktische Fähigkeiten in der Anwendung und Programmierung von Verfahren der Informationsvisualisierung erlangt.		
13. Inhalt:	<p>Die Vorlesung behandelt interaktive Visualisierungen von Datensätzen abstrakter Natur. Dazu gehören beispielsweise quantitative, ordinale und kategorische Daten sowie komplexere Datentypen wie etwa multivariate und relationale. Darüber hinaus wird die Visualisierung von Zeitserien behandelt, die auf jeden Datentyp gleichermaßen zutrifft, deren Visualisierung aber wegen der höheren Komplexität der Daten eine weitergehende Herausforderung darstellt. Anhand von Anwendungsgebieten wie etwa der Bioinformatik, des Software Engineering oder sozialer Netzwerke werden die Vor- und Nachteile der vorgestellten Visualisierungen aufgezeigt. Theoretische und praktische Übungen dienen teilweise der Vorarbeit, aber auch der Vertiefung des behandelten Stoffes. In dieser Vorlesung werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none"> <li>• Visuelle Perzeption und Kognition</li> <li>• Datentypen</li> <li>• Informationsvisualisierungsprinzipien</li> <li>• Designprinzipien nach Tufte</li> <li>• Multivariate Daten und parallele Koordinaten</li> <li>• Hierarchien und Bäume</li> </ul>		

- Graphen und Netzwerke
- Text- und Dokument-Visualisierung
- Diagramme
- Zeitreihen-Visualisierung
- Software-Visualisierung
- Geographische Informationsvisualisierung
- Evaluierung und Benutzerstudien
- Visual Analytics
- Interaktionsprinzipien

14. Literatur:

- Colin Ware. Visual Thinking for Design
- Colin Ware. Information Visualization. Perception for Design
- Edward Tufte. The Visual Display of Quantitative Information
- Robert Spence. Design for Interaction
- Jim Thomas. Illuminating the Path

15. Lehrveranstaltungen und -formen:

556301 Vorlesung und Übung Informationsvisualisierung

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden  
 Selbststudium: 138 Stunden  
**Gesamt: 180 Stunden**

17. Prüfungsnummer/n und -name:

- 55631 Informationsvisualisierung (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0
- 55632 Vorleistung (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

Tafel, Beamer, Matlab-Übungen auf PC

20. Angeboten von:

## Modul: 29630 Konzepte der Programmiersprachen

2. Modulkürzel:	051510312	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Prof.Dr. 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"> <li>• Ghezzi, Jazayeri, Programming Language Concepts, 1987</li> <li>• Sebesta, Robert, Concepts of Programming Languages, 2003</li> </ul>		
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:

---

## 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:	Englisch
8. Modulverantwortlicher:	Prof.Dr. Volker Diekert		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Ulrich Hertrampf</li> <li>• Volker Diekert</li> <li>• Stefan Funke</li> </ul>		
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"> <li>• Bruce Schneier, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C, 1996</li> <li>• Douglas Robert Stinson, Cryptography: Theory and Practice, 1995</li> <li>• Friedrich Ludwig Bauer, Entzifferte Geheimnisse: Methoden und Maximen der Kryptologie, 1995</li> <li>• Johannes Buchmann, Einführung in die Kryptographie, 1999</li> </ul>		
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"> <li>• 29461 Kryptographische Verfahren (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 42870 Message-Basierte Anwendungen

2. Modulkürzel:	052010003	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. 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 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:	021611 Grundlagen der Architektur von Anwendungssystemen, Vorlesung mit Übung, 4,0 SWS		
12. Lernziele:	Am Ende der Veranstaltungen haben die Teilnehmer die Problem der Anwendungsintegration und entsprechende Lösungsansätze verstanden. Message-Orientierte Middleware ist als Basis der häufigsten Lösungen erkannt. Die Architektur von MQ und JMS ist klar. Die wesentlichen Muster, die bei der Integration von Anwendungen zum Einsatz kommen, werden beherrscht. Web Services sind als Integrationstechnologie verstanden.		
13. Inhalt:	Oft werden neue Anwendungen nicht von grund auf neu entwickelt, sondern aus vorhandenen Anwendungen zusammengesetzt. Diese sog. Integration von Anwendungen umfasst den asynchronen Austausch von Nachrichten, meist in verlässlicher Art und Weise ("reliable messaging"). Die Vorlesung für in dieses Gebiet der Unternehmensanwendungs Integration (engl. Enterprise Application Integration EAI) ein. Überblick: Integrationsproblematik; Lose Kopplung; Asynchrone Kommunikation; Messaging Stile (Point-to-Point, Publish-Subscribe, Topics);MQ Netzwerke; JMS; Intergrationsmuster; Kanäle; Message Typen; Routing; Transformation; Endpunkte; Management;		
14. Literatur:	<ul style="list-style-type: none"> <li>• S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, Web Services Platform Architecture, 2005</li> <li>• Zusätzliche Literatur, siehe Webseite zur Veranstaltung</li> </ul>		
15. Lehrveranstaltungen und -formen:	428701 Vorlesung mit Übungen, Message-basierte Anwendungen		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> <li>• 42871 Message-Basierte Anwendungen (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## 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:	Prof.Dr.-Ing. 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 und in mindestens einem Fach aus dem Bereich dem Bereich der Technischen Informatik oder ähnlichen Fächern.		
12. Lernziele:	Studierende beherrschen die praktische Programmierung von Mikrocontrollern und kennen klassische Architekturen. <ul style="list-style-type: none"> <li>• Historische Übersicht</li> <li>• Mikrocontroller-Architekturen</li> <li>• Einsatzgebiete von Mikrocontrollern</li> <li>• Befehlssatz klassischer Microcontroller</li> <li>• Assembler-Programmierung von Mikrocontrollern</li> <li>• C-Programmierung von Mikrocontrollern</li> </ul>		
13. Inhalt:	<p>Als Microcontroller (auch <math>\hat{\mu}</math>Controller, <math>\hat{\mu}</math>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.</p> <p>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.).</p> <p>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<math>\hat{a}</math>, - verfügbar.</p> <p>Aus <a href="http://de.wikipedia.org/wiki/Mikrocontroller">http://de.wikipedia.org/wiki/Mikrocontroller</a></p>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Jörg Wiegmann, Softwareentwicklung in C für Mikroprozessoren und Mikrocontroller: C- Programmierung für Embedded-Systeme, 2009</li> </ul> <p>More literature is named in the lecture</p>		

---

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:	Prof.Dr. Kurt Rothermel		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Kurt Rothermel</li> <li>• Frank Dürr</li> </ul>		
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 bezüglich mobiler Geräte und Systeme sowie drahtloser Kommunikationsverfahren vertieft. Das Lernziel der Vorlesung umfasst das Verstehen der Probleme bei der Verwendung von mobilen Endgeräten, das Entwickeln von Lösungen und die Fähigkeit, mit Experten zu kommunizieren. Die Teilnehmer kennen die Vor- und Nachteile spezifischer drahtloser Kommunikationstechnologien für mobile Geräte. Sie sind sie in der Lage, entsprechende Protokolle für Anwendungen zu verwenden und gegebenenfalls anzupassen. In den Übungen erlangen sie praktische Erfahrung in der Programmierung, Analyse und Leistungsbewertung mobiler Systeme und drahtloser Kommunikationsverfahren und in der Verwendung entsprechender Werkzeuge.</p>		
13. Inhalt:	Grundlagen der drahtlosen Datenübertragung, Medienzugangsverfahren für drahtlose Netze, Lokationsmanagement, drahtlose Weitverkehrsnetze, drahtlose lokale und persönliche Netze, Ad-Hoc-Netze: Vermittlung, Lokationsverwaltung, Mobilität in IP-Netzen, Transportschichtprotokolle für mobile Systeme, Auffinden von Diensten, Mobiler Datenzugriff		
14. Literatur:	Charles E. Perkins: Mobile IP: Design Principles and Practices. 1997 James D. Solomon: Mobile IP: The Internet Unplugged. 1998 Jochen Schiller: Mobile Communications. 2000 Jörg Roth: Mobile Computing: Grundlagen, Technik und Konzepte. 2002 Kian-Lee Tan, Beng-Chin Ooi: Data Dissemination in Wireless Computing Envi-ronments. 2000 Tomasz Imielinski, Henry F. Korth (ed.): Mobile Computing. 1996		
15. Lehrveranstaltungen und -formen:	297201 Vorlesung mit Übung Mobile Computing		
16. Abschätzung Arbeitsaufwand:	Mobile Computing Vorlesung - Präsenzzeit: 21 Stunden - Selbststudium: 69 Stunden Mobile Computing Übungen		

- 
- Präsenzzeit: 21 Stunden
  - Selbststudium: 69 Stunden
- 

17. Prüfungsnummer/n und -name:

- 29721 Mobile Computing (PL), schriftlich oder mündlich, 120 Min.,  
Gewichtung: 1.0
- 29723 Mobile Computing (BSL), schriftlich oder mündlich,  
Gewichtung: 1.0

---

18. Grundlage für ... :

---

19. Medienform: Folien, Tafel

---

20. Angeboten von:

---

## 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:		Prof.Dr.-Ing. Martin Radetzki	
9. Dozenten:			
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:		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"> <li>• Hardware description with VHDL;</li> <li>• Kahn process networks, synchronous data flow networks;</li> <li>• Specification of timing, concurrency, and non-functional aspects;</li> <li>• Object-oriented modelling of embedded systems;</li> <li>• Event-driven simulation;</li> <li>• Modelling levels with emphasis on transaction level modelling;</li> <li>• Application to embedded systems specification;</li> <li>• Integrated hands-on exercises using VHDL and SystemC.</li> </ul>	
14. Literatur:		<ul style="list-style-type: none"> <li>• Lecture Notes "Modelling, Simulation, and Specification".</li> <li>• Jantsch: Modeling Embedded Systems and SoCs Concurrency and Time in Models of Computation. Morgan Kaufman Publishers, 2004.</li> <li>• Black, D.; Donovan, D.: SystemC from the Ground Up. Kluwer Academic Publishers, 2004.</li> <li>• Ashenden, P.J.: The Designer's Guide to VHDL. 2nd edition, Morgan Kaufman Publishers, 2002.</li> <li>• Ashenden, P.J.: The Student's Guide to VHDL. Morgan Kaufman Publishers, 1998.</li> </ul>	
15. Lehrveranstaltungen und -formen:		<ul style="list-style-type: none"> <li>• 297301 Vorlesung Modelling, Simulation, and Specification</li> <li>• 297302 Übung Modelling, Simulation, and Specification</li> </ul>	
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:		<ul style="list-style-type: none"> <li>• 29731 Modelling, Simulation, and Specification (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0</li> <li>• 29732 Modelling, Simulation, and Specification (BSL), schriftlich oder mündlich, Gewichtung: 1.0</li> </ul>	
18. Grundlage für ... :			
19. Medienform:			



20. Angeboten von:

---

## 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 Ankündigung
8. Modulverantwortlicher:	Prof.Dr.-Ing. 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"> <li>• Grundlegende Parallelisierungsansätze: Parallelisierung durch Datenzerlegung, parallele lineare Algebra, etc.</li> <li>• Message Passing Interface</li> <li>• Open MP</li> <li>• C-Programmierung für FPGAs</li> <li>• Graphische Programmierung</li> <li>• GPU-Programmierung</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Thomas Rauber und Gundula Rürger, Multicore: Parallele Programmierung (Informatik Im Fokus), 2007</li> <li>• More literature is named in the lecture</li> </ul>		
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: 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:	Englisch
8. Modulverantwortlicher:	Prof.Dr. 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:	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"> <li>• Attributgrammatiken (Wiederholung)</li> <li>• Speicherorganisation (Speicherverwaltung, Aktivierungsblöcke)</li> <li>• Zwischencode-Erzeugung</li> <li>• Programmanalysen und -Optimierung (Schwerpunkt)</li> <li>• klassische Optimierungen</li> <li>• Lokale und globale Kontrollflussanalyse</li> <li>• Lokale und globale Datenflussanalysen</li> <li>• Dominatoren, Dominatorgrenzen, Kontrollstrukturanalysen</li> <li>• Zeigeranalysen</li> <li>• Seiteneffekt-Analyse</li> <li>• Datenabhängigkeiten, Konfliktanalysen und Registervergabe</li> <li>• SSA-Form und ihre Berechnung</li> <li>• Code-Erzeugung</li> <li>• Implementierung von OOP</li> <li>• Das Laufzeitsystem</li> <li>• Separate Übersetzung</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Aho, Sethi, Ullman, Compilers - Principles, Techniques, and Tools, 1988</li> <li>• Morgan, Robert, Building an Optimizing Compiler, 1998</li> <li>• Muchnick, Steven S., Advanced Compiler Design and Implementation, 1997</li> </ul>		
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, 120 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 Ankündigung
8. Modulverantwortlicher:	Prof.Dr.-Ing. 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"> <li>• James O. Hamblen und Michael D. Furman, Rapid Prototyping of Digital Systems: A Tutorial Approach, 2001</li> <li>• More literature is named in the lecture</li> </ul>		
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:	Prof.Dr. 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.		
13. Inhalt:	<ol style="list-style-type: none"> <li>1) General requirements and terminology of real-time systems</li> <li>2) Deterministic execution: avoiding language-, implementation- and hardware-induced non-determinisms; coping with limited resources; storage estimation and management; execution time estimation</li> <li>3) Fault tolerance: Faults and failure modes, N-version programming, voting, forward and backward recovery</li> <li>4) Simple scheduling regimes: cyclic executives, deadline guarantees</li> <li>5) Parallelism and priority scheduling regimes: processes, threads, tasks; run-time kernels; task management; interrupt handling</li> <li>6) Synchronization and communication: semaphores, critical regions, monitors, protected objects, rendezvous, messaging</li> <li>7) Control of shared resources</li> <li>8) Distributed Systems: basic concepts; major issues</li> </ol>		
14. Literatur:	<ul style="list-style-type: none"> <li>• 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</li> </ul> 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:			

## 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 Ankündigung
8. Modulverantwortlicher:	Prof.Dr.-Ing. 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. Kenntnisse in einem Fach der Technischen Informatik oder einem ähnlichen Fach.		
12. Lernziele:	Die Studierenden kennen Algorithmen, Architekturen und exemplarische Prozessoren zur Echtzeit-Videoverarbeitung.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Einleitung: Analog und Digital Television</li> <li>• Kameras, Bildsensoren und deren Eigenschaften</li> <li>• Image Filtering, Bayer Filter</li> <li>• Motion Analysis</li> <li>• Videokompression</li> <li>• Videokommunikation</li> <li>• Video Processing</li> <li>• Parallele Architekturen, Video Prozessoren und HW-Implementierungen für Real-Time Video Processing Algorithmen</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Roger Clarke und R. J. Clarke von Academic Press Inc, Digital Compression of Still Images and Video (Signal Processing and Its Applications), 1995</li> <li>• More literature is named in the lecture</li> </ul>		
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 Ankündigung
8. Modulverantwortlicher:	Prof.Dr.-Ing. 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"> <li>• Vorstellung der Fallstudie eines Video Processing Systems</li> <li>• Auswahl der Algorithmen des Video Processing Systems</li> <li>• Implementierung und Verifikation der Algorithmen</li> <li>• Architektur-Entwicklung des Video Processing Systems</li> <li>• Performance-Analyse der Achitektur</li> <li>• Implementierung und System-Verifikation</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Roger Clarke und R. J. Clarke von Academic Press Inc, Digital Compression of Still Images and Video (Signal Processing and Its Applications), 1995</li> <li>• More literature is named in the lecture</li> </ul>		
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: 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:	Prof.Dr. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Frank Leymann</li> <li>• Dimka Karastoyanova</li> </ul>		
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 &amp; 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:	428801 Vorlesung mit Übungen, Web Services 1
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 29511 Service Computing (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li></ul>
18. Grundlage für ... :	29520 Ausgewählte Themen des Service Computing
19. Medienform:	Lecture and accompanying exercises
20. Angeboten von:	Architektur von Anwendungssystemen

---

## Modul: 42520 Services und Service Komposition

2. Modulkürzel:	052010008	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Dr.-Ing. Dimka Karastoyanova		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Frank Leymann</li> <li>• Dimka Karastoyanova</li> </ul>		
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:	Vorlesungen des Grundstudiums		
12. Lernziele:	Der Fokus der Vorlesung ist auf Web-basierte Technologien zur Anwendungsintegration vor allem auf der Application Logic Layer. Darüber hinaus werden die Grundprinzipien der Workflow Technologie und ihren Ansatz zur Integration erklärt und gelernt.		
13. Inhalt:	Die „Services und Service Komposition“ Vorlesung erläutert die Architektur von Anwendungen und die Grundprinzipien der Anwendungsintegration. Der Fokus der Vorlesung ist auf Web-basierte Technologien zur Anwendungsintegration und Grundprinzipien der Workflow Technologie und ihren Ansatz zur Integration. Es werden Anwendungsarchitekturstile wie REST und SOA vorgestellt. Entsprechende Technologien und Standards zur Realisierung dieser Architekturstile werden ausführlich diskutiert. Einige Beispiele sind unter anderem Web Services, HTTP, SOAP, WSDL, WS-RF, WS-Policy, WS-Addressing. Die Workflow Technologie wird im Zusammenhang mit dem BPM Lebenszyklus und der entsprechenden Infrastruktur zusammen gefasst. Es wird ein Prozess-Meta-Modell eingeführt und dessen Abbildung auf Prozess-Definition-Sprachen in Detail erläutert. Als Beispiele werden die Sprachen FDL und BPEL gelehrt.		
14. Literatur:	<ul style="list-style-type: none"> <li>• Production Workflow: Concepts and Techniques. Frank Leymann, Dieter Roller. Prentice Hall PTR, 2000</li> <li>• Web Services Platform Architecture: Soap, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More. Sanjiva Weerawarana, Francisco Curbera, Frank Leymann, Donald F. Ferguson, Tony Storey, Prentice Hall International, 2005.</li> </ul>		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> <li>• 425201 Vorlesung Services and Service Compositions</li> <li>• 425202 Übung Services and Service Compositions</li> </ul>		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 48 Stunden Selbststudiumszeit: 132 Stunden		
17. Prüfungsnummer/n und -name:	42521 Services und Service Komposition (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 46760 Theoretische und Methodische Grundlagen des Visual Computing

2. Modulkürzel:	020600012	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.Dr. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> <li>• Marc Alexander Schweitzer</li> <li>• Thomas Müller</li> <li>• Andrés Bruhn</li> </ul>		
10. Zuordnung zum Curriculum in diesem Studiengang:	M.Sc. Softwaretechnik, PO 2012 → Spezialisierungsmodule - MSWT		
11. Empfohlene Voraussetzungen:			
12. Lernziele:	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.		
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. Grundlegende Begriffe der Differentialgeometrie von Kurven und Flächen sowie die Theorie der gewöhnlichen Differentialgleichungen werden vermittelt. Die komplexen Zahlen werden eingeführt bzw. wiederholt und dienen als Grundlage für die Fourier-Theorie. Es wird ein vertieftes Verständnis der kontinuierlichen und diskreten Fourier-Analyse sowie der diskreten Wavelet-Analyse und deren Anwendung in der Bildverarbeitung vermittelt. 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>In dieser Vorlesung werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none"> <li>- Affine und projektive Geometrie: affiner Raum, affine Abbildung, Hauptsatz, orthographische und perspektivische Projektion, projektiver Raum, projektive Abbildung, homogene Koordinaten, 2-/3-Punkt Perspektive, Umsetzung in der Graphikpipeline</li> <li>- Differential- und Integralrechnung: partielle Ableitung, Richtungsableitung, Gradient, Extrema in mehreren Variablen, numerische Ableitung (verrauschter Daten), Kantendetektion, Taylor-</li> </ul>		

Entwicklung in mehreren Variablen, vektorwertige Funktionen, Rotation, Divergenz, Jacobi, Vektorfelder, kritische Punkte, Integralrechnung in mehreren Variablen, Gaußscher Satz, Satz von Stokes, Oberflächenintegral, Bogenlänge, Integraltransformationen

- Differentialgeometrie: Krümmung von Kurven und Flächen, Rotationsflächen
- Gewöhnliche Differentialgleichungen: Existenz und Eindeutigkeit, autonome Systeme, Vektorfelder, analytische Lösung einfacher Systeme, Integralkurven
- Komplexe Zahlen, Quaternionen, Rotationen
- Fourier-Analyse: Zerlegung periodischer Funktionen, Frequenz- und Phasenspektrum, Gibbs, Fourier-Reihe, F-Integral, F-Transformation, Faltung, diskrete Fourier-Transformation, Dirac-delta, Abtasttheorem, Frequenzfilter, Anti-Aliasing, diskrete Filter, Filtermasken, Filterdesign, Anwendungen in der Bildverarbeitung
- Wavelet-Transformation: Haar-Transformation und -Wavelet, Multiresolution-Analyse, Daubechies-Wavelets, Denoising, Bildverarbeitung
- Vollständige Funktionensysteme: Fourierreihen, Kugelflächenfunktionen, allgemeine orthogonale Funktionensysteme, Approximationstheorie, Partition der Eins.
- Punktbasierte Interpolation und Rekonstruktion: Shepard, Moving Least Squares (MLS)
- Epipolargeometrie, Anwendung für Rekonstruktion in der Computer Vision
- Einführung in ein Softwaresystem zur praktischen Umsetzung (z.B. Matlab)

14. Literatur:

- B. Jähne. Digitale Bildverarbeitung (Springer, 2005)
- H. Fischer, H. Kaul. Mathematik für Physiker - Band 1 und 3
- J.S. Walker. A primer on WAVELETS and Their Scientific Applications (Chapman & Hall/CRC, 2008).
- E.J. Stollnitz, T.D. DeRose, D.H. Salesin, „Wavelets for Computer Graphics: A primer“, Computer Graphics and Applications 15, 76-84 (1995)
- M. Oberguggenberger, A. Ostermann. Analysis für Informatiker (Springer, 2009).
- J. Encarnaçã, W. Straßer, R. Klein. Graphische Datenverarbeitung 1 (Oldenburg Verlag, 1996),

15. Lehrveranstaltungen und -formen:

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

17. Prüfungsnummer/n und -name:

- 46761 Theoretische und Methodische Grundlagen des Visual Computing (PL), schriftlich, eventuell mündlich, Gewichtung: 1.0, schriftlich 120 min oder mündlich 30 min
- V Vorleistung (USL-V), schriftlich, eventuell mündlich

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

## 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:	Dr.-Ing. Martin Fuchs		
9. Dozenten:	Martin Fuchs		
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 051900002 Computergraphik		
12. Lernziele:	Die Studierenden kennen die theoretischen Grundlagen des Visual Computing und haben praktische Fähigkeiten in der Programmierung von Techniken für Visual Computing erlangt. Die Studierenden kennen verschiedene Ansätze und Algorithmen, die für Visual Computing einsetzbar sind.		
13. Inhalt:	<p>Diese Vorlesung behandelt die digitale Verarbeitung visueller Information, beginnend mit der Aufzeichnung mit Kameras und aktiven optischen Systemen, über die Aufbereitung zu computergrafischen Modellen und die Wiedergabe mit interaktiven, neuartigen Anzeigesystemen. Sie stützt sich dabei auf Methoden der Computergrafik, algorithmischen Geometrie, Bildbearbeitung aber auch Computer Vision.</p> <p>In dieser Vorlesung werden die folgenden Themen behandelt:</p> <ul style="list-style-type: none"> <li>• Physikalische und mathematische Grundlagen des Visual Computing             <ul style="list-style-type: none"> <li>• Abtastung, Sampling Theorem, Quantisierung</li> <li>• Bildtransformationen und -filterung, AuflösungsPyramiden</li> <li>• Kameramodelle (radiometrisch und optisch)</li> </ul> </li> <li>• Computational Photography, einschließlich High Dynamic Range (HDR)             <ul style="list-style-type: none"> <li>• Integral Imaging / Plenoptische Systeme</li> <li>• Computational Displays</li> <li>• Projektor-Kamera-Systeme</li> </ul> </li> <li>• Szenenvermessung             <ul style="list-style-type: none"> <li>• Active/Passive Stereo</li> <li>• Geometrie aus optischen Eigenschaften (Shape-From-X)</li> <li>• Linearer Lichttransport</li> </ul> </li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• Andrew S. Glassner, Principles of Digital Image Synthesis, 1995</li> <li>• J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics: Principle and Practice, 1990</li> <li>• Jähne, Bernd, Digitale Bildverarbeitung,</li> <li>• Literatur, siehe Webseite zur Veranstaltung</li> </ul>		
15. Lehrveranstaltungen und -formen:	295001 Vorlesung mit Übungen Visual Computing		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudium: 138 Stunden <b>Gesamt: 180 Stunden</b>		

- 
17. Prüfungsnummer/n und -name:
- 29501 Visual Computing (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0, Schriftliche Prüfung von 120 Min. oder mündlichen 30 Min
  - V Vorleistung (USL-V), schriftlich, eventuell mündlich
- 

18. Grundlage für ... :

---

19. Medienform:

---

20. Angeboten von:

---

## Modul: 11330 Visualisierung

2. Modulkürzel:	051900011	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof.Dr. Daniel Weiskopf		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Thomas Ertl</li> <li>• Daniel Weiskopf</li> <li>• Filip Sadlo</li> </ul>		
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>M.Sc. Softwaretechnik, PO 2012          → Spezialisierungsmodule - MSWT</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>• 051900002 Computergraphik</li> <li>• 051900001 Mensch-Computer-Interaktion</li> <li>• 051240005 Numerik und Stochastik.</li> </ul>		
12. Lernziele:	Die Studierenden haben Wissen über Grundlagen, Algorithmen und Datenstrukturen für die Visualisierung sowie praktische Fähigkeiten durch die Arbeit mit Visualisierungssoftware erworben.		
13. Inhalt:	<p>Visualisierung behandelt alle Aspekte, die mit der visuellen Repräsentation von Daten aus wissenschaftlichen Experimenten, Simulationen, medizinischen Scannern, Datenbanken oder ähnlichen Datenquellen gewonnen werden, um zu einem tieferen Verständnis zu gelangen oder eine einfachere Darstellung komplexer Phänomene oder Sachverhalte zu erhalten. Um dieses Ziel zu erreichen, werden zum einen wohlbekannte Techniken aus dem Gebiet der interaktiven Computergraphik, zum anderen auch neu entwickelte Techniken angewendet.</p> <p>Entsprechend werden in dieser Vorlesung folgenden Themen behandelt:</p> <ul style="list-style-type: none"> <li>• Einführung, Historie, Visualisierungspipeline</li> <li>• Datenakquise und -repräsentation (Abtasten, Rekonstruktion, Gitter, Datenstrukturen)</li> <li>• Wahrnehmungsaspekte</li> <li>• Grundlegende Konzepte visueller Abbildungen</li> <li>• Visualisierung von Skalarfeldern (Isoflächenextraktion, Volumenrendering)</li> <li>• Visualisierung von Vektorfelder (Teilchenverfolgung, texturbasierte Methoden, Topologie)</li> <li>• Tensorfelder, Multiattributdaten</li> <li>• Hochdimensionale Daten und Informationsvisualisierung</li> </ul>		
14. Literatur:	<ul style="list-style-type: none"> <li>• C. D. Hansen, C. R. Johnson, The Visualization Handbook, 2005</li> <li>• C. Ware, Information Visualization: Perception for Design, 2004</li> <li>• H. Schumann, W. Müller, Visualisierung: Grundlagen und allgemeine Methoden, 2000</li> </ul>		



---

	<ul style="list-style-type: none"><li>• K. Engel, M. Hadwiger, J. M. Kniss, C. Rezk-Salama, D. Weiskopf, Real-time Volume Graphics, 2006</li></ul>
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"><li>• 113301 Vorlesung Visualisierung</li><li>• 113302 Übungen Visualisierung</li></ul>
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"><li>• 11331 Visualisierung (PL), mündliche Prüfung, 30 Min., Gewichtung: 1.0</li><li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich, 30 Min.</li></ul>
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

---

## Modul: 42900 Workflow Management 1

2. Modulkürzel:	052010006	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Frank Leymann</li> <li>• Dimka Karastoyanova</li> </ul>		
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:	611 Grundlagen der Architektur von Anwendungssystemen, Vorlesung mit Übung, 4,0 SWS		
12. Lernziele:	Am Ende der Veranstaltungen haben die Teilnehmer die wesentlichen Sprachelemente zur Modellierung von Workflows verstanden. Das Konzept der workflowbasierten Anwendungen mit entsprechendem Lebenszyklus ist klar. Die Architektur einer Workflow-Umgebung ist dargestellt. Flow Sprachen (insbesondere BPEL) können verwendet werden. Der graphentheoretische Ansatz von Prozessgraphen als Grundlage von Flow Sprachen ist verstanden. Mechanismen zur Fehler- und Ausnahmebehandlung in Workflows sind klar.		
13. Inhalt:	Workflows sind IT-gestützte Abläufe, die einerseits Geschäftsprozesse unterstützen, andererseits Kompositionen von Anwendungen darstellen. Diese Vorlesung führt in die Grundlagen dieses Gebietes (engl. Business Process Management BPM) ein. <ol style="list-style-type: none"> <li>1. Evolution der Funktionalität von Workflow Technologie</li> <li>2. Business Reengineering (BPM Lifecycle, Werkzeuge,...)</li> <li>3. Architektur von WFMS (Navigator, Executor, Worklist Manager,...)</li> <li>4. Flow Sprachen (FDL, BPEL)</li> <li>5. Prozessmodell Graphen (Mathematisches Metamodel: Syntax, operationelle Semantik)</li> <li>6. Fortgeschrittene Funktionen (Subprozesse, Events, Instanzmodifikationen)</li> <li>7. Zweistufiges Programmieren</li> <li>8. Transaktionsunterstützung in Workflows</li> </ol>		
14. Literatur:	<ul style="list-style-type: none"> <li>• W. van der Aalst, K. van Hee, Workflow Management, 2002</li> </ul>		
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"> <li>• 42901 Workflow Management 1 (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

## Modul: 42910 Workflow Management 2

2. Modulkürzel:	052010007	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Frank Leymann		
9. Dozenten:	<ul style="list-style-type: none"> <li>• Frank Leymann</li> <li>• Dimka Karastoyanova</li> </ul>		
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:	<p>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.</p>		
13. Inhalt:	<p>In der Vorlesung werden fortgeschrittene Themen des Workflowmanagement vorgestellt. Aktuelle Entwicklungen aus dem Forschungsumfeld und der Industrie auf dem Gebiet werden diskutiert.</p> <p>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</p>		
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"> <li>• 42911 Workflow Management 2 (PL), schriftlich oder mündlich, 120 Min., Gewichtung: 1.0</li> <li>• V Vorleistung (USL-V), schriftlich, eventuell mündlich</li> </ul>		
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.Dr. 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"> <li>• Rechtsstruktur des Internet</li> <li>• Haftung von Internetanbietern (Caching, Hosting, Foren, Links, Anschlussinhaber etc.)</li> <li>• Internetstrafrecht ( Viren, Hacking, Domain grabbing, ...)</li> <li>• Datenschutz und Globale Netzöffentlichkeit</li> <li>• Meine Domain, deine Domain - Namensrecht im Internet</li> <li>• Rechtsprobleme des eCommerce (Verbraucherschutz, Internetauktionen, Signaturregelung)</li> <li>• EGovernment (Internetwahlen, eCampaigning, elektronisches Rathaus etc.)</li> </ul>		
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:	[pord.modulcode]	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.Dr. Stefan Wagner		
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 Teilnehmer haben einen Überblick über die für Software relevanten Rechtsgebiete und können konkrete Problemstellungen einschätzen.		
13. Inhalt:	<ul style="list-style-type: none"> <li>• Urheberrechtlicher Schutz von Computerprogrammen</li> <li>• Patentierung von Computerprogrammen</li> <li>• Elektronischer Geschäftsverkehr</li> <li>• Kauf/Miete/Werkvertrag</li> <li>• Datenschutz und personenbezogene Daten</li> </ul>		
14. Literatur:	IT- und Computerrecht. Deutscher Taschenbuchverlag, 2010		
15. Lehrveranstaltungen und -formen:	428401 Vorlesung Software-Recht		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 28 Stunden, Nachbearbeitungszeit: 56 Stunden		
17. Prüfungsnummer/n und -name:	42841 Software-Recht (USL), schriftlich, eventuell mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			
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:

---