



Universität Stuttgart

Modulhandbuch
Studiengang Bachelor of Arts (Kombination) Informatik
Prüfungsordnung: 2009
Nebenfach

Wintersemester 2012/13
Stand: 10. Oktober 2012

Universität Stuttgart
Keplerstr. 7
70174 Stuttgart

Kontaktpersonen:

Studiendekan/in:	Univ.-Prof.Dr. Daniel Weiskopf Institut für Visualisierung und Interaktive Systeme Tel.: E-Mail: daniel.weiskopf@vis.uni-stuttgart.de
Studiengangsmanager/in:	Corinna Vehlow Visualisierungsinstitut der Universität Stuttgart Tel.: E-Mail: corinna.vehlow@visus.uni-stuttgart.de
Prüfungsausschussvorsitzende/r:	Univ.-Prof.Dr. Otto Eggenberger Institut für Rechnergestützte Ingenieursysteme Tel.: E-Mail: otto.eggenberger@iris.uni-stuttgart.de
Fachstudienberater/in:	Bernhard Schmitz Institut für Visualisierung und Interaktive Systeme Tel.: E-Mail: Bernhard.Schmitz@vis.uni-stuttgart.de

Inhaltsverzeichnis

Präambel	4
100 Module im Nebenfach	5
42480 Ausgewählte Kapitel des Wissenschaftlichen Rechnens	6
23530 Automaten und Formale Sprachen	8
12060 Datenstrukturen und Algorithmen	9
14360 Einführung in die Technische Informatik	11
42420 High Performance Computing	13
320 Katalog ISG	15
330 Katalog ISW	16
42460 Numerische Simulation	17
10260 Programmierkurs	19
10280 Programmierung und Software-Entwicklung	20

Präambel

Die Studien- und Prüfungsordnung für den Bachelorstudiengang **Informatik beschreibt den** Aufbau des Studiums und die Organisation der Prüfungen. Sie stellt das Regelwerk und die Rechtsgrundlage für eine einheitliche Handhabung des Studienablaufs und der Bewertung der Studien- und Prüfungsleistungen dar. Sie wendet sich dabei sowohl an die Studierenden als auch an die Prüfenden sowie an die entsprechenden Organe der Universität Stuttgart.

Aus Gründen der sprachlichen Vereinfachung wird grundsätzlich nur die grammatikalisch männliche Form für Personen verwendet. Sinngemäß ist stets auch die entsprechende weibliche Form gemeint.

100 Module im Nebenfach

Zugeordnete Module:	10260	Programmierkurs
	10280	Programmierung und Software-Entwicklung
	12060	Datenstrukturen und Algorithmen
	14360	Einführung in die Technische Informatik
	23530	Automaten und Formale Sprachen
	320	Katalog ISG
	330	Katalog ISW
	42420	High Performance Computing
	42460	Numerische Simulation
	42480	Ausgewählte Kapitel des Wissenschaftlichen Rechnens

Modul: 42480 Ausgewählte Kapitel des Wissenschaftlichen Rechnens

2. Modulkürzel:	051240030	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof.Dr. Marc Alexander Schweitzer		
9. Dozenten:	<ul style="list-style-type: none"> • Stefan Zimmer • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Informatik, PO 2009 → Ergänzungsmodule → Katalog ISG 1-3</p> <p>B.Sc. Informatik, PO 2012 → Ergänzungsmodule → Katalog ISG 1-3</p> <p>BA (Komb) Informatik, PO 2009 → Module im Nebenfach</p> <p>BA (Komb) Informatik, PO 2009 → Module im Nebenfach → Katalog ISG</p> <p>BA (Komb) Informatik, PO 2009 → Module im Nebenfach → Katalog ISW</p>		
11. Empfohlene Voraussetzungen:	<p>080300100 Mathematik für Informatiker und Softwaretechniker und 051240005 Numerische und Stochastische Grundlagen der Informatik bzw. 051240006 Einführung in die Numerik und Stochastik für Softwaretechniker 051240020 Grundlagen des wissenschaftlichen Rechnens</p>		
12. Lernziele:	Die Teilnehmer kennen ausgewählte aktuelle Forschungsthemen des wissenschaftlichen Rechnens und können mit der zugehörigen Primärliteratur arbeiten.		
13. Inhalt:	Aktuelle weiterführende Themen des wissenschaftlichen Rechnens, z.B. adaptive Finite Elemente, Fehlerschätzer, hierarchische Basen und dünne Gitter, robuste Multilevellöser, p-Version und Spektralverfahren.		
14. Literatur:	<p>Primärliteratur zu den behandelten Themen:</p> <ul style="list-style-type: none"> • Bungartz/Griebel: Sparse Grids; Acta Numerica, Volume 13, p. 147-269 • Quarteroni/Valli: Numerical approximation of partial differential equations • Quarteroni: Numerical models for differential problems 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 424801 Vorlesung Ausgewählte Kapitel des Wissenschaftlichen Rechnens • 424802 Übung Ausgewählte Kapitel des Wissenschaftlichen Rechnens 		
16. Abschätzung Arbeitsaufwand:	<p>Präsenzzeit: 42 Stunden Selbststudiumszeit: 138 Stunden</p>		
17. Prüfungsnummer/n und -name:	42481 Ausgewählte Kapitel des Wissenschaftlichen Rechnens (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 23530 Automaten und Formale Sprachen

2. Modulkürzel:	050420007	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. Volker Diekert		
9. Dozenten:	<ul style="list-style-type: none"> • Ulrich Hertrampf • Volker Diekert 		
10. Zuordnung zum Curriculum in diesem Studiengang:	BA (Komb) Informatik, PO 2009, 4. Semester → Module im Nebenfach		
11. Empfohlene Voraussetzungen:	Grundkenntnisse über Logik und Diskrete Strukturen, Mathematik		
12. Lernziele:	Die Studierenden beherrschen wichtige theoretische Grundlagen der Informatik, insbesondere die Theorie und Algorithmik endlicher Automaten. Hierzu gehört das Kennenlernen, Einordnen und Trennung der Chomskyschen Sprachklassen.		
13. Inhalt:	Deterministische- bzw. nichtdeterministische endliche Automaten, reguläre Ausdrücke, Minimierung endlicher Automaten, Iterationslemmata für reguläre und kontextfreie Sprachen, Normalformen, Kellerautomaten, Lösen des Wortproblems kontextfreier Sprachen mit dem CYK-Algorithmus, linear beschränkte Automaten, kontextsensitive Grammatiken, Typ 0-Grammatiken und Turingmaschinen.		
14. Literatur:	Uwe Schöning, Theoretische Informatik - kurzgefasst, 1999		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 235301 Vorlesung Automaten und Formale Sprachen • 235302 Gruppenübungen Automaten und Formale Sprachen 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 23531 Automaten und Formale Sprachen (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0 • V Vorleistung (USL-V), schriftlich, eventuell mündlich 		
18. Grundlage für ... :	<ul style="list-style-type: none"> • 10020 Algorithmik • 14910 Berechenbarkeit und Komplexität 		
19. Medienform:			
20. Angeboten von:			

Modul: 12060 Datenstrukturen und Algorithmen

2. Modulkürzel:	051510005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Thomas Ertl		
9. Dozenten:	Stefan Funke		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 2. Semester → Basismodule B.Sc. Informatik, PO 2012, 2. Semester → Basismodule BA (Komb) Informatik, PO 2009, 2. Semester → Module im Nebenfach		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 051520005 Programmierung und Software-Entwicklung 		
12. Lernziele:	<p>Die Studierenden kennen nach engagierter Mitarbeit in dieser Veranstaltung diverse zentrale Algorithmen auf geeigneten Datenstrukturen, die für eine effiziente Nutzung von Computern unverzichtbar sind. Sie können am Ende zu gängigen Problemen geeignete programmiersprachliche Lösungen angeben und diese in einer konkreten Programmiersprache formulieren.</p> <p>Konkret:</p> <ul style="list-style-type: none"> • Kenntnis der Eigenschaften elementarer und häufig benötigter Algorithmen • Verständnis für die Auswirkungen theoretischer und tatsächlicher Komplexität • Erweiterung der Kompetenz im Entwurf und Verstehen von Algorithmen und der zugehörigen Datenstrukturen • Erste Begegnung mit nebenläufigen Algorithmen; sowohl „originär“ parallel, als auch parallelisierte Versionen bereits vorgestellter sequentieller Algorithmen 		
13. Inhalt:	<ul style="list-style-type: none"> • Vorgehensweise bei der Entwicklung und Implementierung von Algorithmen • Komplexität und Effizienz von Algorithmen, O-Notation • Wahl der Datenstrukturen; Listen, Bäume, Graphen; deren Definitionen, deren Datenstrukturen • diverse interne und externe Such- und Sortierverfahren (z.B. Linear-, Binär-, Interpolationssuche, AVL-, B-Bäume, internes und externes Hashing, mehrere langsame Sortierungen, Heap-, Quick-, Bucket-, Mergesort) • diverse Graphenalgorithmen (DFS, BFS, Besuchssequenzen, topol. Traversierung, Zusammenhangskomponenten, minimale Spannbäume, Dijkstra-, Floyd- kürzeste Wege) • Algorithmen auf Mengen und Relationen (transitive Hüllen, Warshall) • Korrektheitsbegriff und -formalismen; Spezifikation und Implementierung • Einige parallele und parallelisierte Algorithmen • einfache Elemente paralleler Programmierung, soweit für obiges notwendig 		
14. Literatur:	<ul style="list-style-type: none"> • Appelrath H.J., Ludewig. J., Skriptum Informatik, 1999 • Sedgewick, R., Algorithms in C, 1998 		

15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 120601 Vorlesung Datenstrukturen und Algorithmen• 120602 Übung Datenstrukturen und Algorithmen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 63 Stunden Nachbearbeitungszeit: 207 Stunden
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none">• 12061 Datenstrukturen und Algorithmen (PL), schriftlich, eventuell mündlich, Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein. Die genauen Details der Übungsleistungen und Ihrer Anrechnung werden zu Beginn der Lehrveranstaltung bekannt gegeben.• V Vorleistung (USL-V), schriftlich, eventuell mündlich
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

Modul: 14360 Einführung in die Technische Informatik

2. Modulkürzel:	051400105	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof.Dr. Otto Eggenberger		
9. Dozenten:	Otto Eggenberger		
10. Zuordnung zum Curriculum in diesem Studiengang:	BA (Komb) Informatik, PO 2009, 3. Semester → Module im Nebenfach		
11. Empfohlene Voraussetzungen:	Keine		
12. Lernziele:	Der/die Studierende kennt die grundlegende Funktionsweise eines Computers, versteht die elektrotechnischen Grundlagen und Technologien und kann einfache digitale Schaltungen analysieren, entwerfen und optimieren.		
13. Inhalt:	<p>Grundlegende Funktionsweise eines Computers</p> <ul style="list-style-type: none"> • Informationsdarstellung • Zahlendarstellung und Codes • Digitale Grundbausteine • Logische Funktionen, Speicherelemente • Befehlsausführung, Programmablauf <p>Elektrotechnische Grundlagen</p> <ul style="list-style-type: none"> • Physikalische Grundbegriffe • Elektrische Spannung, elektrischer Strom • Elektrische Netzwerke • Halbleiterbauelemente • Digitale Grundschaltungen <p>Digitale Schaltungen</p> <ul style="list-style-type: none"> • Schaltnetzwerke • Boolesche Algebra und Schaltalgebra • Darstellung und Minimierung von Schaltfunktionen • Rückkopplung, Zustandsbegriff • Automaten und sequentielle Netzwerke • Digitale Standardschaltungen • Entwurfsmethodik 		
14. Literatur:	Dirk W. Hoffman: Grundlagen der technischen Informatik, Hanser, 2007 Bernd Becker, Rolf Drechsler, Paul Molitor: Technische Informatik, Pearson Studium, 2005 Jörg Keller, Wolfgang J. Paul: Hardware Design, Teubner, 3. Aufl. 2005		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 143601 Vorlesung Einführung in die Technische Informatik • 143602 Gruppenübungen Einführung in die Technische Informatik 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	42 Stunden	
	Nachbearbeitungszeit:	126 Stunden	
17. Prüfungsnummer/n und -name:	14361 Einführung in die Technische Informatik (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von:

Modul: 42420 High Performance Computing

2. Modulkürzel:	051240040	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof.Dr. Marc Alexander Schweitzer		
9. Dozenten:	<ul style="list-style-type: none"> • Martin Bernreuther • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Informatik, PO 2009 → Ergänzungsmodule → Katalog ISG 1-3</p> <p>B.Sc. Informatik, PO 2012 → Ergänzungsmodule → Katalog ISG 1-3</p> <p>BA (Komb) Informatik, PO 2009 → Module im Nebenfach</p> <p>BA (Komb) Informatik, PO 2009 → Module im Nebenfach → Katalog ISG</p> <p>BA (Komb) Informatik, PO 2009 → Module im Nebenfach → Katalog ISW</p>		
11. Empfohlene Voraussetzungen:	<p>080300100 Mathematik für Informatiker und Softwaretechniker und 051240005 Numerische und Stochastische Grundlagen der Informatik bzw. 051240006 Einführung in die Numerik und Stochastik für Softwaretechniker</p>		
12. Lernziele:	<p>Fähigkeit, parallele Algorithmen auf unterschiedlichen parallelen Plattformen mit Hilfe geeigneter algorithmischer Modelle zu bewerten. Kenntnis verschiedener Programmiermodelle für Parallelrechner mit verteiltem und gemeinsamem Speicher.</p> <p>Fähigkeit, auch fortgeschrittene Implementierungsaufgaben aus dem Bereich des Höchstleistungsrechnens auf Basis ausgewählter Programmiermodelle zu bewältigen.</p>		
13. Inhalt:	<p>Die Vorlesung beschäftigt sich mit den Grundlagen paralleler Programmierung und paralleler Algorithmen speziell im Hinblick auf die Anwendungsbereiche Wissenschaftliches Rechnen und High Performance Computing.</p> <p>Verwandte Fragestellungen aus dem Bereich der Theorie (parallele Modelle und parallele Komplexität, etc.) sowie aus der Rechnertechnik (parallele Architekturen) werden begleitend diskutiert.</p> <p>Nach einer allgemeinen Einführung (Klassifizierung von Parallelrechnern, Ebenen von Parallelität, Performance und Architekturen, etc.), werden die Grundlagen paralleler Programme eingeführt (Notation/Syntax, Synchronisation und Kommunikation, Design paralleler Programme, etc.). Sowohl die Programmierung auf Systemen mit gemeinsamem Speicher als auch auf Systemen mit verteiltem Speicher werden besprochen. Dabei wird jeweils mindestens ein geeignetes Programmiermodell (z.B. OpenMP, MPI, CUDA) vertieft behandelt.</p> <p>Aus dem Bereich des High Performance Computing werden begleitend klassische Algorithmen und Implementierungstechniken als Beispiele</p>		

behandelt, z.B. parallele Algorithmen aus der linearen Algebra (Matrixmultiplikation, etc. oder einfache Verfahren zur Lösung partieller Differentialgleichungen). Zusätzlich können Themen wie Lastverteilung und Lastbalancierung (Grundlagen, Algorithmen zur Partitionierung und Lastbalancierung, etc.) vorgestellt werden.

14. Literatur:	<ul style="list-style-type: none">• T. Rauber, G. Runger: „Parallele Programmierung“, 2. Aufl., Springer 2007; (in English: T. Rauber, G. Runger: „Parallel Programming: for Multicore and Cluster Systems“, Springer 2010)• K.A. Berman, J.L. Paul: "Sequential and Parallel Algorithms", PWS Publishing Company, 1997• B. Chapman, G. Jost, R. van der Pas: "Using OpenMP - Portable Shared Memory Parallel Programming", MIT Press, 2008• W. Gropp, E. Lusk, und R. Thakur: "Using MPI-2: Advanced Features of the Message-Passing Interface", das Buch ist auch in deutscher ubersetzung erhaltlich.• D. Kirk, W.-M. Hwu Programming Massively Parallel Processors
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 424201 Vorlesung High Performance Computing• 424202 ubung High Performance Computing
16. Abschatzung Arbeitsaufwand:	Prsenzzeit: 42 Stunden Selbststudiumszeit: 138 Stunden
17. Prufungsnummer/n und -name:	42421 High Performance Computing (PL), schriftlich oder mundlich, 90 Min., Gewichtung: 1.0
18. Grundlage fur ... :	
19. Medienform:	
20. Angeboten von:	

320 Katalog ISG

330 Katalog ISW

Modul: 42460 Numerische Simulation

2. Modulkürzel:	051240060	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof.Dr. Marc Alexander Schweitzer		
9. Dozenten:	<ul style="list-style-type: none"> • Stefan Zimmer • Marc Alexander Schweitzer 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Informatik, PO 2009 → Ergänzungsmodule → Katalog ISW 1-3</p> <p>B.Sc. Informatik, PO 2012 → Ergänzungsmodule → Katalog ISW 1-3</p> <p>BA (Komb) Informatik, PO 2009 → Module im Nebenfach</p> <p>BA (Komb) Informatik, PO 2009 → Module im Nebenfach → Katalog ISG</p> <p>BA (Komb) Informatik, PO 2009 → Module im Nebenfach → Katalog ISW</p>		
11. Empfohlene Voraussetzungen:	080300100 Mathematik für Informatiker und Softwaretechniker und 051240005 Numerische und Stochastische Grundlagen der Informatik bzw. 051240006 Einführung in die Numerik und Stochastik für Softwaretechniker 051240020 Grundlagen des wissenschaftlichen Rechnens		
12. Lernziele:	Fähigkeit zur Implementierung numerischer Methoden und Entwicklung und Umsetzung geeigneter Datenstrukturen.		
13. Inhalt:	Strukturmechanik, Strömungsmechanik, Finite Elemente, Finite Differenzen, Verallgemeinerte Finite Elemente		
14. Literatur:	<ul style="list-style-type: none"> • Griebel, Dornseifer, Neunhoeffler: Numerical simulation in fluid dynamics : a practical introduction; SIAM, 1998 / Numerische Simulation in der Strömungsmechanik; Vieweg 1995 • Griebel, Knappek, Zumbusch, Caglar: Numerische Simulation in der Moleküldynamik : Numerik, Algorithmen, Parallelisierung, Anwendungen; Springer 2004 • Braess: Finite Elemente : Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie; Springer, 2007 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 424601 Vorlesung Numerische Simulation • 424602 Übung Numerische Simulation 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Selbststudiumszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	42461 Numerische Simulation (PL), schriftlich oder mündlich, 90 Min., Gewichtung: 1.0		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von:

Modul: 10260 Programmierkurs

2. Modulkürzel:	051520010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	3.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Univ.-Prof.Dr. Stefan Wagner	
9. Dozenten:		<ul style="list-style-type: none"> • N. N. • Ivan Bogicevic 	
10. Zuordnung zum Curriculum in diesem Studiengang:		B.Sc. Informatik, PO 2009, 1. Semester → Basismodule BA (Komb) Informatik, PO 2009, 1. Semester → Module im Nebenfach	
11. Empfohlene Voraussetzungen:		Keine	
12. Lernziele:		Selbstständiges Erstellen von Programmen und Lösung von Programmieraufgaben in einer vorgegebenen Programmiersprache wie Java.	
13. Inhalt:		<p>Der Programmierkurs ergänzt die Vorlesung Programmierung und Software-Entwicklung (PSE). Die Teilnehmer erlernen eine weitere Programmiersprache (Java). Ihre Merkmale, Syntax und Semantik, werden denen der in PSE gelehrt Sprache gegenübergestellt. Praktische Übungen bereiten die Teilnehmer auf die Bearbeitung der Schein-Aufgabe vor.</p> <p>Die Lehrveranstaltung findet in zwei Varianten statt. Die Teilnahme richtet sich nach dem Studiengang:</p> <p>S. Riexinger:</p> <ul style="list-style-type: none"> • BSc. Informatik • BA (Komb) Informatik • BSc. Maschinelle Sprachverarbeitung <p>H. Röder:</p> <ul style="list-style-type: none"> • BSc. Softwaretechnik • BSc. Wirtschaftsinformatik • BSc. Technikpädagogik • MSc. Technikpädagogik 	
14. Literatur:			
15. Lehrveranstaltungen und -formen:		102601 Übung Programmierkurs	
16. Abschätzung Arbeitsaufwand:		Präsenzzeit: 21 Stunden Nachbearbeitungszeit: 69 Stunden	
17. Prüfungsnummer/n und -name:		10261 Programmierkurs (USL), Sonstiges, Gewichtung: 0.0, Übungsschein - Scheinkriterien werden zu Beginn der Veranstaltung angekündigt.	
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 10280 Programmierung und Software-Entwicklung

2. Modulkürzel:	051520005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Univ.-Prof.Dr. Stefan Wagner		
9. Dozenten:	Bernhard Mitschang		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 1. Semester → Basismodule B.Sc. Informatik, PO 2012, 1. Semester → Basismodule BA (Komb) Informatik, PO 2009, 1. Semester → Module im Nebenfach		
11. Empfohlene Voraussetzungen:	Keine. Teilnahme an einem Mathematik Vorkurs wird empfohlen.		
12. Lernziele:	Die Teilnehmer haben die wichtigsten Konzepte einer höheren Programmiersprache und ihrer Verwendung verstanden und sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen) zu analysieren und selbst zu konzipieren und zu implementieren. Sie kennen die Möglichkeiten, Daten- und Ablaufstrukturen zu entwerfen, zu beschreiben und zu codieren. Sie haben die Abstraktionskonzepte moderner Programmiersprachen verstanden. Sie kennen die Techniken und Notationen zur Definition kontextfreier Programmiersprachen und können damit arbeiten.		
13. Inhalt:	<ul style="list-style-type: none"> • Die Programmiersprache Java und die virtuelle Maschine • Objekte, Klassen, Schnittstellen, Blöcke, Programmstrukturen, Kontrakte • Klassenmodellierung mit der UML • Objekterzeugung und -ausführung • Boolesche Logik • Verzweigungen, Schleifen, Routinen, Abstraktionen, Modularisierung, Variablen, Zuweisungen • Rechner, Hardware • Syntaxdarstellungen • Übersicht über Programmiersprachen und -werkzeuge • Grundlegende Datenstrukturen und Algorithmen • Vererbung, Polymorphe • Semantik • Programmierung graphischer Oberflächen • Übergang zum Software Engineering 		
14. Literatur:	<ul style="list-style-type: none"> • Appelrath, Hans-Jürgen und Ludewig, Jochen, "Skriptum Informatik - eine konventionelle Einführung", Verlag der Fachvereine Zürich und B.G. Teubner Stuttgart, 4. Auflage 1999 • Meyer, Bertrand, "Touch of Class", Springer-Verlag, 2009 • Savitch, Walter, "Java. An Introduction to Problem Solving and Programming", Pearson, 6. Auflage, 2012 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 102801 Vorlesung Programmierung und Softwareentwicklung • 102802 Übung Programmierung und Softwareentwicklung 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit:	63 Stunden	

Vor-/Nachbearbeitungszeit: 187 Stunden

Prüfungsvorbereitung: 20 Stunden

17. Prüfungsnummer/n und -name:

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

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:
