



Universität Stuttgart

Modulhandbuch
Studiengang Lehramt an Gymnasien (GymPO I) Informatik
Prüfungsordnung: 2010
Hauptfach

Wintersemester 2012/13
Stand: 09. 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

200 Pflichtmodule	4
27620 Algorithmen und Berechenbarkeit für Lehramt	5
12060 Datenstrukturen und Algorithmen	6
14360 Einführung in die Technische Informatik	8
10190 Mathematik für Informatiker und Softwaretechniker	10
27630 Praktische Informatik für Lehramt	12
10280 Programmierung und Software-Entwicklung	13
10320 Seminar-INF 1	15
27640 Systemkonzepte und -programmierung für Lehramt	17
10940 Theoretische Grundlagen der Informatik	18
 300 Wahlmodule	 20
10140 Advanced Processor Architecture	21
10080 Datenbanken und Informationssysteme	23
10100 Grundlagen der Graphischen Ingenieursysteme	25
10150 Grundlagen des Compilerbaus und der Programmiersprachen	27
 400 Fachdidaktikmodule	 29
34050 Gestaltung von Lehr- / Lernprozessen im Informatik-Unterricht, Projekt	30
34060 Grundlagen der Fachdidaktik Informatik	31
 3000 Zwischenprüfung	 32
12060 Datenstrukturen und Algorithmen	33
10280 Programmierung und Software-Entwicklung	35
10940 Theoretische Grundlagen der Informatik	37

200 Pflichtmodule

Zugeordnete Module:	10190	Mathematik für Informatiker und Softwaretechniker
	10280	Programmierung und Software-Entwicklung
	10320	Seminar-INF 1
	10940	Theoretische Grundlagen der Informatik
	12060	Datenstrukturen und Algorithmen
	14360	Einführung in die Technische Informatik
	27620	Algorithmen und Berechenbarkeit für Lehramt
	27630	Praktische Informatik für Lehramt
	27640	Systemkonzepte und -programmierung für Lehramt

Modul: 27620 Algorithmen und Berechenbarkeit für Lehramt

2. Modulkürzel:	050420021	5. Moduldauer:	1 Semester
3. Leistungspunkte:	7.0 LP	6. Turnus:	jedes 2. Semester, WiSe
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:			
11. Empfohlene Voraussetzungen:	Vorlesungen aus dem 1. und 2. Semester		
12. Lernziele:			
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:	276201 Vorlesung mit Übungen Algorithmen und Berechenbarkeit		
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	27621 Algorithmen und Berechenbarkeit für Lehramt (LBP), schriftlich, eventuell mündlich, Gewichtung: 1.0		
18. Grundlage für ... :			
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:	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.
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: 10190 Mathematik für Informatiker und Softwaretechniker

2. Modulkürzel:	080300100	5. Moduldauer:	2 Semester
3. Leistungspunkte:	18.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	12.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Apl. Prof.Dr. Wolfgang Rump		
9. Dozenten:	Wolfgang Rump		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 1. Semester → Basismodule B.Sc. Informatik, PO 2012, 1. Semester → Basismodule		
11. Empfohlene Voraussetzungen:	Keine, die Teilnahme an einem Mathematik Vorkurs wird empfohlen.		
12. Lernziele:	Die Studierenden haben die mathematischen Grundlagen für die Studiengänge Informatik bzw. Softwaretechnik erarbeitet und den selbständigen und kreativen Umgang mit den mathematischen Stoffgebieten gelernt.		
13. Inhalt:	1. Semester: <ul style="list-style-type: none"> • Grundlagen (Aussagenlogik, Mengen, Relationen, Abbildungen, Zahlenmengen, Grundbegriffe der Algebra) • Lineare Algebra (Vektorräume, lineare Abbildungen, Matrizen, Determinanten, lineare Gleichungssysteme, Eigenwerte, Normalformen, Hauptachsentransformation, Skalarprodukte) • Analysis (Konvergenz, Zahlenfolgen und Zahlenreihen, stetige Abbildungen, Folgen und Reihen von Funktionen, spezielle Funktionen) 2. Semester: <ul style="list-style-type: none"> • Differential- und Integralrechnung (Funktionen einer und mehrerer Variablen, Ableitungen, Taylorentwicklungen, Extremwerte, Integration, Anwendungen) • Gewöhnliche Differentialgleichungen (elementar lösbare Differentialgleichungen, Existenz und Eindeutigkeit von Lösungen) 		
14. Literatur:	<ul style="list-style-type: none"> • Anna Sändig, Mathematik, Vorlesungskripte , SS 2007 • D. Hachenberger, Mathematik für Informatiker, 2005 • M. Brill, Mathematik für Informatiker, 2001 • P.Hartmann, Mathematik für Informatiker, 2002 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101901 Vorlesung Mathematik 1 für Informatik und Softwaretechnik • 101902 Übung Mathematik 1 für Informatik und Softwaretechnik • 101903 Vorlesung Mathematik 2 für Informatik und Softwaretechnik • 101904 Übung Mathematik 2 für Informatik und Softwaretechnik 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 126 Stunden Nachbearbeitungszeit: 414 Stunden		
17. Prüfungsnummer/n und -name:	10191 Mathematik für Informatiker und Softwaretechniker (PL), schriftliche Prüfung, 120 Min., Gewichtung: 1.0, Ein Übungsschein aus den beiden Veranstaltungen, jeweils im 1. oder 2. Fachsemester zu erwerben		
18. Grundlage für ... :			

19. Medienform:

20. Angeboten von:

Modul: 27630 Praktische Informatik für Lehramt

2. Modulkürzel:	050420031	5. Moduldauer:	1 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Prof.Dr. Stefan Funke	
9. Dozenten:			
10. Zuordnung zum Curriculum in diesem Studiengang:			
11. Empfohlene Voraussetzungen:			
12. Lernziele:			
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 276301 Vorlesung Praktische Informatik für Lehramt (Teil A) • 276302 Vorlesung mit Übungen Modellierung (Teil B1) • 276303 Vorlesung mit Übungen Mensch-Computer-Interaktion (Teil B2) 		
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 27631 Praktische Informatik für Lehramt B1 (PL), schriftlich, eventuell mündlich, Gewichtung: 1.0 • 27632 Praktische Informatik für Lehramt B2 (PL), schriftlich, eventuell mündlich, Gewichtung: 1.0 • 27633 Praktische Informatik für Lehramt, Projektschein (USL), schriftlich, eventuell mündlich, Gewichtung: 1.0 		
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

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 10320 Seminar-INF 1

2. Modulkürzel:	050420095	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:	<ul style="list-style-type: none"> • Dozenten der Informatik • Dozenten der Anorganischen Chemie 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 5. Semester → Schlüsselqualifikationen fachaffin		
11. Empfohlene Voraussetzungen:	Basismodule der Informatik, darüber hinaus variabel: Je nach dem gewählten Seminarthema können Vorkenntnisse aus weiteren Vorlesungen benötigt werden.		
12. Lernziele:	<p>Die Studierenden können sich mit wissenschaftlicher Originalliteratur auseinandersetzen, deren Kernaussagen rezipieren und sich ein spezielles Thema überwiegend im Selbststudium erarbeiten. Sie sind fähig relevante Daten zu sammeln und zu interpretieren und ihre Erkenntnisse einem Fach- und Laienpublikum verständlich zu präsentieren und auf Fragen aus dem Publikum angemessen und sachgerecht zu reagieren. Sie haben gelernt, sich mit einem wissenschaftlichen Thema über einen längeren Zeitraum hinweg auseinander zu setzen und eigenständig aktuelle Hintergrundinformation zu beschaffen. Sie haben generische Kompetenzen erworben, etwa aktiv an einer wissenschaftlichen Diskussion zu einem vorher bekannten Thema teilzunehmen und durch Fragen an den Vortragenden ihr Verständnis zu erweitern. Sie können eine Diskussion leiten und moderieren und sind befähigt, ihre Ergebnisse den Seminarteilnehmern vorzustellen und mit Hilfe moderner Präsentationstechniken zu visualisieren.</p>		
13. Inhalt:	<p>Variabel: Es werden Seminare zu diversen häufig aktuellen Themen angeboten.</p> <p>Das Seminar INF kann in der Informatik oder in einem affinen Fach durchgeführt werden, wie etwa Computerlinguistik, Elektrotechnik, Mathematik oder Wirtschaftswissenschaften. Welche Seminare zugelassen sind, entscheidet die Studienkommission. Zugelassene Seminare werden typischer Weise durch Aushang bekannt gegeben. Die Seminare sind in Größe und Inhalt so gestaltet, dass die generischen Kompetenzen (Schlüsselqualifikationen) der Studierenden entwickelt werden.</p>		
14. Literatur:	<ul style="list-style-type: none"> • Literatur, die begleitende Literatur wird in der Veranstaltung und im Web bekannt gegeben. 		
15. Lehrveranstaltungen und -formen:	103201 Seminar		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 21 Stunden Nachbearbeitungszeit: 69 Stunden Gesamt: 90 Stunden		
17. Prüfungsnummer/n und -name:	10321 Seminar-INF 1 (LBP), Sonstiges, Gewichtung: 1.0, Scheinkriterien sind in der Regel ein Vortrag, eine schriftliche Ausarbeitung, sowie die aktive Mitarbeit während der Seminarveranstaltung.		

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 27640 Systemkonzepte und -programmierung für Lehramt

2. Modulkürzel:	051200095	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch

8. Modulverantwortlicher:

9. Dozenten:

10. Zuordnung zum Curriculum in diesem Studiengang:

11. Empfohlene Voraussetzungen:

12. Lernziele:

13. Inhalt:

14. Literatur:

15. Lehrveranstaltungen und -formen: 276401 Vorlesung mit Übungen Systemkonzepte und -programmierung

16. Abschätzung Arbeitsaufwand:

17. Prüfungsnummer/n und -name: 27641 Systemkonzepte und -programmierung für Lehramt (LBP), schriftlich, eventuell mündlich, Gewichtung: 1.0

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 10940 Theoretische Grundlagen der Informatik

2. Modulkürzel:	050420005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Volker Diekert		
9. Dozenten:	<ul style="list-style-type: none"> • Ulrich Hertrampf • Volker Diekert 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 1. Semester → Basismodule B.Sc. Informatik, PO 2012, 1. Semester → Basismodule		
11. Empfohlene Voraussetzungen:	Keine		
12. Lernziele:	<ul style="list-style-type: none"> • Logik und Diskrete Strukturen: <p>Die Studierenden haben die grundsätzlichen Kenntnisse in Logik und Diskreter Mathematik erworben, wie sie in den weiteren Grundvorlesungen der Informatik in verschiedenen Bereichen benötigt werden.</p> <ul style="list-style-type: none"> • Automaten und Formale Sprachen: <p>Die Studierenden beherrschen wichtige theoretische Grundlagen der Informatik, insbesondere die Theorie und Algorithmik endlicher Automaten. Hierzu gehört das Kennenlernen, Einordnung und Trennung der Chomskyschen Sprachklassen.</p>		
13. Inhalt:	<ul style="list-style-type: none"> • Logik und Diskrete Strukturen: <p>Einführung in die Aussagenlogik; formale Sprache; Semantik (Wahrheitswerte); Syntax (Axiome und Schlussregeln); Normalformen; Hornformeln; aussagenlogische Resolution; Korrektheit und Vollständigkeit für die Aussagenlogik; Einführung in die Prädikatenlogik 1. Stufe; formale Sprache; Semantik und Syntax; Normalformen; Herbrand-Theorie; prädikatenlogische Resolution; Kombinatorik, Graphen, elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, RSA-Verfahren.</p> <ul style="list-style-type: none"> • Automaten und Formale Sprachen: <p>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.</p>		
14. Literatur:	<ul style="list-style-type: none"> • John Hopcroft, Jeffrey Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988 • Uwe Schöning, Theoretische Informatik - kurzgefasst, 1999 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 109401 Vorlesung Logik und Diskrete Strukturen • 109402 Übung Logik und Diskrete Strukturen • 109403 Vorlesung Automaten und Formale Sprachen 		

-
- 109404 Übung Automaten und Formale Sprachen

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 84 Stunden
Nachbearbeitungszeit: 276 Stunden

17. Prüfungsnummer/n und -name:

10941 Theoretische Grundlagen der Informatik (PL), schriftliche Prüfung, Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

300 Wahlmodule

Zugeordnete Module: 10080 Datenbanken und Informationssysteme
 10100 Grundlagen der Graphischen Ingenieursysteme
 10140 Advanced Processor Architecture
 10150 Grundlagen des Compilerbaus und der Programmiersprachen

Modul: 10140 Advanced Processor Architecture

2. Modulkürzel:	051700010	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Englisch
8. Modulverantwortlicher:	Prof.Dr. Hans-Joachim Wunderlich		
9. Dozenten:	<ul style="list-style-type: none"> • Hans-Joachim Wunderlich • Stefan Holst 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Informatik, PO 2009, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3</p> <p>B.Sc. Informatik, PO 2012, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3</p> <p>BA (Komb) Informatik, PO 2009, 5. Semester → Module im Nebenfach → Katalog ISG</p> <p>BA (Komb) Informatik, PO 2009, 5. Semester → Module im Nebenfach → Katalog ISW</p>		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • 051700005 Rechnerorganisation 		
12. Lernziele:	<p>Good understanding of the basic concepts used in modern CPUs and computing systems. Awareness of the challenges in modern processor design and the reasoning behind current and future design trends.</p>		
13. Inhalt:	<p>Classic topics in computer architecture as hardware/software interface are discussed as well as more advanced topics which include:</p> <ul style="list-style-type: none"> • Technology basics: Design patterns, fabrication, yield, test and reliability, cost and quality, scaling. • Performance: Frequency and instructions per clock cycle, performance analysis and optimization. • Power dissipation: Analysis and optimization of power and performance, power and scaling. • Computer arithmetic: Efficient hardware for basic arithmetic, implementation of exponential, logarithm and trigonometric functions, floating point arithmetic and standards, arithmetic pipelines and filter, real-world floating point implementations like the Cell SPE or SPARC. • Instruction parallelism: Super scalar computing, static and dynamic scheduling, out-of-order execution, VLIW-processors, multithreading • Parallel architectures: Shared memory and message passing, multi-core processors, multi-core systems on a chip and emerging many-core technologies found in current graphic accelerators • Memory hierarchy: Memory technology and cache design. • Fault tolerance for single processors and multi processor systems 		

14. Literatur:	<ul style="list-style-type: none">• I. Koren, Computer Arithmetic Algorithms, 2001• J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 2012• S. Iman, M. Pedram, Logic Synthesis for Low Power VLSI Designs, 1997
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 101401 Vorlesung Grundlagen der Rechnerarchitektur• 101402 Übung Grundlagen der Rechnerarchitektur
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10141 Advanced Processor Architecture (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

Modul: 10080 Datenbanken und Informationssysteme

2. Modulkürzel:	051200025	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.-Ing. Bernhard Mitschang		
9. Dozenten:	<ul style="list-style-type: none"> • Bernhard Mitschang • Holger Schwarz 		
10. Zuordnung zum Curriculum in diesem Studiengang:	<p>B.Sc. Informatik, PO 2009, 5. Semester</p> <ul style="list-style-type: none"> → Ergänzungsmodule → Katalog ISG 1-3 <p>BA (Komb) Informatik, PO 2009, 5. Semester</p> <ul style="list-style-type: none"> → Module im Nebenfach → Katalog ISG 		
11. Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> • Modul 052010001 Modellierung 		
12. Lernziele:	Die Studierenden haben das erforderliche Verständnis und die Kenntnisse für die Implementierung von Datenbanksystemen erworben.		
13. Inhalt:	<p>Die Vorlesung "Datenbanken und Informationssysteme" ist als Einstiegsveranstaltung in das Vertiefungsgebiet Datenbanksysteme konzipiert. Aufbauend auf dem Inhalt der Vorlesung "Modellierung" werden insbesondere Entwurfs- und Realisierungsaspekte von Datenbanksystemen betrachtet. Die Entwicklung, Installation und Administration von Datenbanksystemen bestimmen hier sowohl Stoffauswahl als auch Detaillierungsgrad.</p> <p>Als Grundlage für alle weiteren Betrachtungen wird ein Schichtenmodell zur Beschreibung eines allgemeinen Datenbanksystems vorgestellt. Darauf aufbauend werden die einzelnen Systemschichten im Detail diskutiert, die dort zu realisierenden Komponenten betrachtet sowie die jeweils vorherrschenden Algorithmen beschrieben und bewertet. Im einzelnen werden folgende Aspekte vertieft:</p> <ul style="list-style-type: none"> • Anwendungsprogrammierschnittstelle • Externspeicherverwaltung • DBS-Pufferverwaltung • Speicherungsstrukturen und Zugriffspfadstrukturen • Anfrageverarbeitung und Anfrageoptimierung • Transaktionsverarbeitung • Synchronisation • Logging und Recovery 		
14. Literatur:	<ul style="list-style-type: none"> • A. Kemper, A. Eickler, Datenbanksysteme - Eine Einführung, 2004 • T. Härder, E. Rahm, Datenbanksysteme - Konzepte und Techniken der Implementierung, 2001 • H. Garcia-Molina, J. D. Ullman, J. Widom, Database Systems. The Complete Book, 2003 		

- R. Elmasri, S. Navathe, Fundamentals of Database Systems, 2003

15. Lehrveranstaltungen und -formen:

- 100801 Vorlesung Datenbanken und Informationssysteme
- 100802 Übung Datenbanken und Informationssysteme

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 42 Stunden
Nachbearbeitungszeit: 138 Stunden

Gesamt: 180 Stunden

17. Prüfungsnummer/n und -name:

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 10100 Grundlagen der Graphischen Ingenieursysteme

2. Modulkürzel:	051400005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Dieter Roller		
9. Dozenten:	Dieter Roller		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 B.Sc. Informatik, PO 2012, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3 BA (Komb) Informatik, PO 2009, 4. Semester → Module im Nebenfach → Katalog ISG BA (Komb) Informatik, PO 2009, 4. Semester → Module im Nebenfach → Katalog ISW		
11. Empfohlene Voraussetzungen:	Grundstudium		
12. Lernziele:	<ul style="list-style-type: none"> • Kenntnis und Verständnis von Modellen bei der Produktentwicklung • Grundkenntnisse über die wichtigsten Modellarten, Algorithmen und Datenstrukturen und Techniken für den Datenaustausch 		
13. Inhalt:	Inhalte: <ul style="list-style-type: none"> • Anforderungen an CAD-Systeme • zweidimensionale Modelle • dreidimensionale Modelle • interaktive Modellerstellung • Einführung in die Modifikationstechnik u. parametrische Modellierung • Methoden zur Modellmodifikation • Grundlagen der parametrischen Modellierung • Ansätze und Verfahren zur parametrischen Variantenerzeugung • Ausgewählte Anwendungsbeispiele • Überblick über weitergehende Modellieransätze • Datenverwaltung in CAD 		
14. Literatur:	<ul style="list-style-type: none"> • D. Roller, CAD - Effiziente Anpassungs- und Variantenkonstruktion, Springer-Verlag • Literatur, siehe Webseite zur Veranstaltung 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 101001 Vorlesung Grundlagen der Graphischen Ingenieursysteme • 101002 Übung Grundlagen der Graphischen Ingenieursysteme 		
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden		
17. Prüfungsnummer/n und -name:	10101 Grundlagen der Graphischen Ingenieursysteme (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0		
18. Grundlage für ... :			
19. Medienform:			

20. Angeboten von:

Modul: 10150 Grundlagen des Compilerbaus und der Programmiersprachen

2. Modulkürzel:	051510015	5. Moduldauer:	1 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Prof.Dr. Erhard Plödereder	
9. Dozenten:		Erhard Plödereder	
10. Zuordnung zum Curriculum in diesem Studiengang:		<p>B.Sc. Informatik, PO 2009, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3</p> <p>B.Sc. Informatik, PO 2012, 3. Semester → Ergänzungsmodule → Katalog ISG 1-3</p> <p>BA (Komb) Informatik, PO 2009, 5. Semester → Module im Nebenfach → Katalog ISG</p> <p>BA (Komb) Informatik, PO 2009, 5. Semester → Module im Nebenfach → Katalog ISW</p>	
11. Empfohlene Voraussetzungen:		<p>Grundkenntnisse aus den Einführungsvorlesungen des Informatikgrundstudiums, sowie einige Erfahrungen mit Programmierung. Vorkenntnisse über formale Sprachen sind vorteilhaft, aber nicht zwingend.</p>	
12. Lernziele:		<p>Die Studierenden haben die Grundkenntnisse erlangt, die zur effizienten Verwendung von Lexer- und Parser-Generatoren zur Analyse von Eingabetexten nötig sind. Sie verstehen die grundlegende Funktionsweise mehrerer Parse-Verfahren und kennen deren grammatikalischen Einschränkungen. Sie haben gelernt, die Fehlermeldungen aus diesen Generatoren und den Compilern oder Interpretern richtig einzuordnen. Ferner haben sie durch Betrachtung der Implementierungsmodelle typischer Programmiersprachenkonstrukte Verständnis für das Ausführungsverhalten und für typische, gefährliche Fehlerquellen in Anwendungsprogrammen erlangt.</p>	
13. Inhalt:		<p>Compilerarchitekturen im Überblick; lexikalische und syntaktische Analyse von Texten mit formaler Grammatik, insb. von Programmiersprachen. Lexikalische Analyse: endliche Automaten und ihre Implementierung; Syntaxanalyse: diverse Parser-Strategien, ihre Implementierung und Eigenschaften. Methoden der automatischen Generierung von Analysatoren aus Spezifikationen der Grammatiken. Fehlererkennung und -behandlung. Analyse der statischen Semantik: Grundbegriffe und elementare Methoden. Eigenschaften von Programmiersprachen; Realisierung der Laufzeitsemantik prozeduraler Programmiersprachen aus Benutzersicht, insbesondere Implementierungsmodelle der Speicherverwaltung und der Unterprogrammaufrufe. Vermeidung typischer Fehlerquellen und überraschender Probleme in Anwendungsprogrammen.</p> <p>(Nach SS14 wird sich der programmiersprachliche Teil ändern.)</p>	
14. Literatur:		<ul style="list-style-type: none"> • Aho, Sethi, Ullman, Compilers - Principles, Techniques, and Tools, 1988 • Wilhelm, Maurer, Uebersetzerbau, 1997 	

15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none">• 101501 Vorlesung Grundlagen des Compilerbaus und der Programmiersprachen• 101502 Übung Grundlagen des Compilerbaus und der Programmiersprachen
16. Abschätzung Arbeitsaufwand:	Präsenzzeit: 42 Stunden Nachbearbeitungszeit: 138 Stunden
17. Prüfungsnummer/n und -name:	10151 Grundlagen des Compilerbaus und der Programmiersprachen (PL), schriftliche Prüfung, 60 Min., Gewichtung: 1.0
18. Grundlage für ... :	
19. Medienform:	
20. Angeboten von:	

400 Fachdidaktikmodule

Zugeordnete Module: 34050 Gestaltung von Lehr- / Lernprozessen im Informatik-Unterricht, Projekt
 34060 Grundlagen der Fachdidaktik Informatik

Modul: 34050 Gestaltung von Lehr- / Lernprozessen im Informatik-Unterricht, Projekt

2. Modulkürzel:	101010072	5. Moduldauer:	1 Semester
3. Leistungspunkte:	4.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	2.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:		Apl. Prof.Dr. Ulrich Hertrampf	
9. Dozenten:			
10. Zuordnung zum Curriculum in diesem Studiengang:			
11. Empfohlene Voraussetzungen:			
12. Lernziele:			
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:	340501	Seminar Gestaltung von Lehr- / Lernprozessen im Informatik-Unterricht, Projekt	
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	34051	Gestaltung von Lehr- / Lernprozessen im Informatik-Unterricht, Projekt (PL), schriftlich, eventuell mündlich, Gewichtung: 1.0	
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

Modul: 34060 Grundlagen der Fachdidaktik Informatik

2. Modulkürzel:	101010062	5. Moduldauer:	2 Semester
3. Leistungspunkte:	6.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	4.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Reinhold Nickolaus		
9. Dozenten:			
10. Zuordnung zum Curriculum in diesem Studiengang:			
11. Empfohlene Voraussetzungen:			
12. Lernziele:			
13. Inhalt:			
14. Literatur:			
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 340601 Vorlesung Einführung in die Didaktik der Informatik • 340602 Seminar Vertiefungen zur Einführung in die Didaktik der Informatik 		
16. Abschätzung Arbeitsaufwand:			
17. Prüfungsnummer/n und -name:	<ul style="list-style-type: none"> • 34061 Grundlagen der Fachdidaktik Informatik (PL), schriftlich, eventuell mündlich, Gewichtung: 1.0 • 34062 Grundlagen der Fachdidaktik Informatik, Ausarbeitung incl. Präsentation (USL), schriftlich, eventuell mündlich, Gewichtung: 1.0 		
18. Grundlage für ... :			
19. Medienform:			
20. Angeboten von:			

3000 Zwischenprüfung

Zugeordnete Module: 10280 Programmierung und Software-Entwicklung
 10940 Theoretische Grundlagen der Informatik
 12060 Datenstrukturen und Algorithmen

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:	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.
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

18. Grundlage für ... :

19. Medienform:

20. Angeboten von:

Modul: 10940 Theoretische Grundlagen der Informatik

2. Modulkürzel:	050420005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Prof.Dr. Volker Diekert		
9. Dozenten:	<ul style="list-style-type: none"> • Ulrich Hertrampf • Volker Diekert 		
10. Zuordnung zum Curriculum in diesem Studiengang:	B.Sc. Informatik, PO 2009, 1. Semester → Basismodule B.Sc. Informatik, PO 2012, 1. Semester → Basismodule		
11. Empfohlene Voraussetzungen:	Keine		
12. Lernziele:	<ul style="list-style-type: none"> • Logik und Diskrete Strukturen: <p>Die Studierenden haben die grundsätzlichen Kenntnisse in Logik und Diskreter Mathematik erworben, wie sie in den weiteren Grundvorlesungen der Informatik in verschiedenen Bereichen benötigt werden.</p> <ul style="list-style-type: none"> • Automaten und Formale Sprachen: <p>Die Studierenden beherrschen wichtige theoretische Grundlagen der Informatik, insbesondere die Theorie und Algorithmik endlicher Automaten. Hierzu gehört das Kennenlernen, Einordnung und Trennung der Chomskyschen Sprachklassen.</p>		
13. Inhalt:	<ul style="list-style-type: none"> • Logik und Diskrete Strukturen: <p>Einführung in die Aussagenlogik; formale Sprache; Semantik (Wahrheitswerte); Syntax (Axiome und Schlussregeln); Normalformen; Hornformeln; aussagenlogische Resolution; Korrektheit und Vollständigkeit für die Aussagenlogik; Einführung in die Prädikatenlogik 1. Stufe; formale Sprache; Semantik und Syntax; Normalformen; Herbrand-Theorie; prädikatenlogische Resolution; Kombinatorik, Graphen, elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, RSA-Verfahren.</p> <ul style="list-style-type: none"> • Automaten und Formale Sprachen: <p>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.</p>		
14. Literatur:	<ul style="list-style-type: none"> • John Hopcroft, Jeffrey Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988 • Uwe Schöning, Theoretische Informatik - kurzgefasst, 1999 		
15. Lehrveranstaltungen und -formen:	<ul style="list-style-type: none"> • 109401 Vorlesung Logik und Diskrete Strukturen • 109402 Übung Logik und Diskrete Strukturen • 109403 Vorlesung Automaten und Formale Sprachen 		

- 109404 Übung Automaten und Formale Sprachen

16. Abschätzung Arbeitsaufwand:

Präsenzzeit: 84 Stunden
Nachbearbeitungszeit: 276 Stunden

17. Prüfungsnummer/n und -name:

10941 Theoretische Grundlagen der Informatik (PL), schriftliche Prüfung, Gewichtung: 1.0, Prüfungsvorleistung: Übungsschein

18. Grundlage für ... :

19. Medienform:

20. Angeboten von: