

Modulhandbuch

Bachelor-Studiengang
Informatik

an der
Johannes Gutenberg-Universität Mainz

Stand: * >|cã Áã Á ÚÁG€€ì €€

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-01	Pflicht

Modultitel	Mathematik für Informatiker				
Empfohlen für	1.- 2. Semester				
Verantwortlich	Institut für Mathematik				
Dauer	2 Semester				
Modulturnus	beginnt in jedem Wintersemester				
Lehrformen		Titel	SWS	Präsenz- stunden	Selbst- studium
	Vorlesung	Mathematik für Informatiker I + II	2 x 6	180	180
	Übung	Mathematik für Informatiker I + II	2 x 4	120	240
	Praktikum	Mathematisches Praktikum	2	40	90
Arbeitsaufwand	34 CP = 850 Arbeitsstunden (Workload)				
Verwendbarkeit	B.Sc. Informatik				
Ziele	Der Modul vermittelt die für Informatiker wichtigsten Grundlagen der Mathematik sowie die damit verbundene Kenntnis der unterschiedlichsten mathematischen Strukturen. Dabei wird das Abstraktionsvermögen geschult und ein tieferes Verständnis für das Formalisieren von Vorgängen und Problemen erzielt. Die vermittelten Beweistechniken erhöhen die Fähigkeit, Probleme gezielt auf das Wesentliche zu reduzieren und dann einer Lösung zuzuführen.				
Inhalt	<p>> Vorlesung und Übungen:</p> <ul style="list-style-type: none"> • Aussagen, Beweise, Mengen, Relationen, Abbildungen, vollständige Induktion; • Kombinatorik und diskrete Wahrscheinlichkeitsrechnung: Binomialkoeffizienten, unabhängige Ereignisse, bedingte Wahrscheinlichkeit, Zufallsvariablen, Erwartungswerte und Varianz; • Graphentheorie: Eulersche und Hamiltonsche Graphen, Bäume, gewichtete Graphen, ebene und plättbare Graphen, Färbungen auf Graphen; • Elementare Zahlentheorie: Äquivalenzklassen, Teilbarkeit und Division mit Rest, Primzahlen, Rechnen modulo n, Primzahltests; • Kryptographie: Klassische Verfahren und RSA-Algorithmus; • Algebraische Strukturen: Gruppen, Ringe, Polynomringe, endliche Körper, komplexe Zahlen, Lösungsformeln für Gleichungen, Vektorräume, Erzeugendensysteme und Dimension, Homomorphismen; • Erzeugende Funktionen und lineare Rekursionsgleichungen; • Lineare Abbildungen und Matrizen, lineare Gleichungssysteme und Determinante, Eigenwerte, charakteristisches Polynom, Minimalpolynom, Jordansche Normalform; • Codierungstheorie: Lineare Codes, perfekte Codes, zyklische Codes; • Reelle Analysis: Reelle Zahlen, Folgen und Grenzwerte, Reihen, (gleichmäßige) Stetigkeit, 				

	<p>Differentiation und Integration, trigonometrische Funktionen, Logarithmus und Exponentialfunktion, Funktionenfolgen und gleichmäßige Konvergenz, Funktionenreihen, analytische Funktionen und Taylor-Reihe;</p> <ul style="list-style-type: none"> • Numerische Verfahren: Nullstellenbestimmung, Interpolation, Quadratur. <p>> Praktikum:</p> <ul style="list-style-type: none"> • Deterministische und probabilistische Algorithmen der diskreten Mathematik und algorithmischen Zahlentheorie; • Programmieren von konkreten Algorithmen in einer Programmiersprache wie Java, C, Python oder einer Skriptsprache eines Computeralgebrasystems wie MAPLE oder MUPAD.
Teilnahmevoraussetzungen	keine
Literaturangabe	keine
Vergabe von Leistungspunkten	<p>Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen (Prüfungsvorleistung), Klausuren (Prüfungsleistung), und am Praktikum (Prüfungsleistung).</p> <p>Die Modulnote ergibt sich in CP-gewichteter Form aus den Klausurnoten und der Beurteilung der im Zuge des Praktikums erarbeiteten Programme.</p>
Prüfungsformen und Leistungen	<p>Modulabschlussprüfung:</p> <ul style="list-style-type: none"> > Klausuren zu den Vorlesungen/Übungen: 4 x 180 Minuten > Erstellung und Erläuterung von Programmen im Praktikum

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-02	Pflicht

Modultitel	Einführung in die Programmierung				
Empfohlen für	1. Semester				
Verantwortlich	Institut für Informatik				
Dauer	2 Semester				
Modulturnus	Jedes Wintersemester				
		Titel	SWS	Präsenzstd.	Selbststudium
	Vorlesung	Einführung in die Programmierung (EIP)	2	30	60
	Übung	Einführung in die Programmierung (EIP)	2	30	60
	Vorlesung	Einführung in die Softwareentwicklung (EIS)	2	30	60
	Übung	Einführung in die Softwareentwicklung (EIS)	2	30	60
	Vorlesung	Programmiersprachen (PS)	2	30	60
	Übung	Programmiersprachen (PS)	2	30	60
Arbeitsaufwand	3x6 CP = 540 Arbeitsstunden				
Verwendbarkeit	B.Sc. Informatik				
Ziele	<p>EIP: Beherrschung einer objektorientierten Programmiersprache; Grundfertigkeiten zum Algorithmen- und Software-Entwurf</p> <p>Softwaresysteme werden i. Allg. heute nach objektorientierten Ansätzen entwickelt. Das Modul führt in die Grundlagen der Entwicklung objektorientierter Systeme ein und erprobt diese am praktischen Beispiel.</p> <p>EIS: Ausgehend vom Einsatz objektorientierter Modellierungsmethoden zur Beschreibung von Softwaresystemen (hier UML) wird die Realisierung, die Dokumentation und der Test des Systems vermittelt. Die Realisierung erfolgt in einer objektorientierten Programmiersprache (hier Java) unter Verwendung relevanter Bibliotheken für Standardtypen (Collections) und graphischer Benutzungsschnittstellen (Swing).</p> <p>Der praktische Anteil der Veranstaltung wird durch Standard-Software-Entwicklungswerkzeuge (z.Z. Eclipse, SVN, JavaDoc, JUnit) unterstützt.</p> <p>PS: Kennenlernen der logischen und funktionalen Programmierparadigmen; Vertiefung des Zeiger- und Adresskonzepts in der Programmiersprache C</p>				
Inhalt	<p>EIP: Variablen-Begriff, Kontrollstrukturen, Felder, Unterprogramme, Rekursion, Klassenkonzept; Algorithmen zum Suchen und Sortieren, etc.; Software-</p>				

	<p>Entwicklungszyklus</p> <p>EIS: Prozessmodelle der Softwareentwicklung;</p> <ul style="list-style-type: none"> • Objektorientierung (Grundlagen der Objektorientierung, Vorgehen zur objektorientierten Softwareentwicklung); • UML als Modellierungsmittel (Objektdiagramme, Klassendiagramme); Objektorientierte Implementierung; • Testen (Testgrundlagen, Testfälle und Teststrategien, Testen mit JUnit); Ausnahmebehandlung; • abstrakte Datenstrukturen (Java-Collections); GUI Entwicklung mit Swing <p>PS: Die Programmiersprachen Prolog, Scheme und C</p>
Teilnahmevoraussetzungen	keine
Literaturangabe	<p>Lehrbücher zum Programmieren in Java;</p> <p>Helmut Balzer: Objektorientierte Programmierung mit Java 5. Spektrum</p> <p>Lehrbücher der Informatik, 2005.</p> <p>Reinhard Schiedermeier: Programmieren mit Java, Eine methodische Einführung, Pearson, 2005</p> <p>Guido Krüger: Handbuch der Java-Programmierung, 4. Auflage, Addison-Wesley, 2006.</p> <p>Chris Rupp, Jürgen Hahn, Stefan Queins, Mario Jeckle, Barbara Zengler: UML 2 glasklar Praxiswissen für die UML -Modellierung und Zertifizierung, Hanser, 2005.</p> <p>Lehrbücher wie Henning/Vogelsang „Programmiersprachen“</p>
Vergabe von Leistungspunkten	Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen (Prüfungsvorleistung) und Klausur (Prüfungsleistung). Die Klausurnoten ergeben die Modulnote.
Prüfungsformen und Leistungen	Modulabschlussprüfung: je Vorlesung eine Klausuren von 120 – 180 Minuten Dauer

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-03	Pflicht

Modultitel	Theoretische Grundlagen der Informatik				
Empfohlen für	2. Semester				
Verantwortlich	Institut für Informatik				
Dauer	2 Semester				
Modulturnus	Jedes Sommersemester				
Lehrformen		Titel	SWS	Präsenzstd.	Selbststudium
	Vorlesung	Theoretische Grundlagen der Informatik I	2	30	60
	Übung	Theoretische Grundlagen der Informatik I	2	30	60
	Vorlesung	Theoretische Grundlagen der Informatik II	2	30	60
	Übung	Theoretische Grundlagen der Informatik II	2	30	60
Arbeitsaufwand	12 CP = 360 Arbeitsstunden				
Verwendbarkeit	B.Sc. Informatik				
Ziele	Vermittlung der theoretischen Grundlagen der Informatik, Beherrschung der formalen Konzepte				
Inhalt	<p>Teil 1: Formale Sprachen und Grammatiken, endliche Automaten und Kellerautomaten, Chomsky-Hierarchie</p> <p>Teil 2: Berechenbarkeitstheorie (Turing-Maschinen, Unentscheidbarkeit, Reduktion) und Komplexitätstheorie (Aufwandsabschätzung, Klassen P und NP, NP-Vollständigkeit)</p>				
Teilnahmevoraussetzungen	Mathematik für Informatiker I, wünschenswert „Einführung in die Programmierung“				
Literaturangabe	Vergleichbar zu Hopcroft/Motwani/Ullman: „Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie“				
Vergabe von Leistungspunkten	Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen (Prüfungsvorleistung) und Klausur (Prüfungsleistung). Die Klausurnote ist die Modulnote.				
Prüfungsformen und Leistungen	Modulabschlussprüfung Klausur 180 Minuten				

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-04	Pflicht

Modultitel	Software Technik				
Empfohlen für	3. Semester				
Verantwortlich	AG Softwaretechnik				
Dauer	1 Semester				
Modulturnus	jedes Wintersemester				
Lehrformen		Titel	SWS	Präsenzstd.	Selbststudium
	Vorlesung	Software Engineering	2	30	45
	Übung		2	30	75
	Praktikum		2	80	10
Arbeitsaufwand	9 ECTS = 270 Arbeitsstunden (Workload), davon 6 ECTS = 180 Arbeitsstunden Vorlesung und 3 ECTS = 90 Arbeitsstunden Praktikum				
Verwendbarkeit	Pflichtmodul im B.Sc.Informatik				
Ziele	<p>Softwaretechnik (Software-Engineering) ist die Teildisziplin der Informatik, welche sich mit der Entwicklung und Anwendung von Prinzipien, Methoden und Werkzeugen zur Erstellung, zum Betrieb und zur Wartung von großen Softwaresystemen befasst. Ziel der Veranstaltung ist es, entlang der zentralen Tätigkeiten zur Entwicklung von Softwaresystemen einen Überblick über diese Prinzipien, Methoden und Werkzeuge zu geben. Diese Veranstaltung soll die Teilnehmer in die Lage versetzen, die Vorgehensweisen und Hilfsmittel der Softwaretechnik in den verschiedenen Phasen der Software-Entwicklung und -Wartung einschätzen und anwenden zu können.</p>				
Inhalt	<p>Inhalte der Vorlesung:</p> <ol style="list-style-type: none"> 1. Einführung <ol style="list-style-type: none"> 1.1 Zielsetzung 1.2 Literatur 1.3 Motivation (Softwarefehler, Probleme der Softwareentwicklung) 1.4 Ingenieursdisziplin "Softwaretechnik" 1.5 Prinzipien der Softwaretechnik 2. Software-Entwicklungsprozess <ol style="list-style-type: none"> 2.1 Prozess und Aktivitäten 2.2 Prozessmodelle der Software-Entwicklung (incl. Unified Process, eXtreme Programming, V-Modell XT) 3. Modellierung <ol style="list-style-type: none"> 3.1 Modelle, Modellbegriff, 3.2 objektorientierte Modellierung 3.3 Unified Modeling Language 4. Anforderung erheben <ol style="list-style-type: none"> 4.1 Aktivität "Anforderung erheben" 4.2 Anforderungen 4.3 Lastenheft und Anforderungsdefinition 4.4 Modellierungsmittel 				

	<ul style="list-style-type: none"> 5. dynamische Modellierungsmittel <ul style="list-style-type: none"> 5.1 Aktivitätsdiagramme 5.2 Datenflussdiagramme 5.3 Statechartes (Statecharts) 6. Softwaresystem entwerfen <ul style="list-style-type: none"> 6.1 Aktivität "Softwaresystem entwerfen" 6.2 Modellierungsmittel 6.3 Software Architektur 6.4 Software Spezifikation (textuell, algebraisch, konstruktiv) 7. Softwaresystem implementieren <ul style="list-style-type: none"> 7.1 Aktivität "Softwaresystem implementieren" 7.2 Programmierrichtlinien 7.3 Entwurfs-Muster 8. Qualität sichern <ul style="list-style-type: none"> 8.1 Aktivität "Qualität sichern" 8.2 Inspizieren 8.3 Testen 9. Projekt planen <ul style="list-style-type: none"> 9.1 Aktivität "Projekt planen" 9.2 Projektplan erstellen 9.3 Team planen 9.4 Projektkalkulation erstellen 10. Softwaresystem weiterentwickeln <p>Inhalte des Praktikums: Projektaufgabe zur Anwendung und Vertiefung des in der Vorlesung erworbenen Wissens</p>
Teilnahme- voraussetzungen	Vorlesung Einführung in die Softwareentwicklung
Literaturangabe	<p>Ian Sommerville: Software Engineering, Addison-Wesley, Wokingham, 8. Auflage, 2006.</p> <p>Helmut Balzert: Lehrbuch der Software-Technik, Band 1: Software-Entwicklung, Spektrum, Heidelberg, 2. Auflage, 2001, Band 2: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung, Spektrum, Heidelberg, 1998.</p> <p>Chris Rupp, Jürgen Hahn, Stefan Queins, Mario Jeckle, Barbara Zengler: UML 2 glasklar. 2. Auflage, Hanser, 2005.</p>
Vergabe von Leistungspunkten	Klausur zur Vorlesung, im Praktikum unbenotete Prüfungsleistung
Prüfungsformen und Leistungen	<p>Vorlesung: Klausur mind. 120 min, erfolgreiche Teilnahme an den Übungen</p> <p>Praktikum: Projektarbeit, Präsentation, Ausarbeitung</p>

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-05	Pflicht

Modultitel	Technische Informatik				
Empfohlen für	1. Semester				
Verantwortlich	Institut für Informatik/ Zentrum für Datenverarbeitung				
Dauer	1 Semester				
Modulturnus	jedes Wintersemester				
Lehrformen		Titel	SWS	Präsenzstd.	Selbststudium
	Vorlesung	Technische Informatik	3	45	60
	Übung	Technische Informatik	1	15	45
	Praktikum				
Arbeitsaufwand	6 CP = 165 Arbeitsstunden (Workload)				
Verwendbarkeit	B.Sc. Informatik				
Ziele	<p>Der Modul vermittelt einen Einblick in die Architektur und technische Realisierung von Rechnersystemen.</p> <p>Die Studierenden sollen dabei ein Verständnis für die Abläufe in Rechnersystemen entwickeln und lernen, welche Komponenten den Ablauf von Programmen besonders beeinflussen, um Engpässe und Optimierungsmöglichkeiten bei Programmen besser zu verstehen.</p>				
Inhalt	<ul style="list-style-type: none"> - Grundfunktionen elektronischer Schaltkreise, zugehörige Schaltlogik, Aufbau komplexerer Schaltungen und technologische Aspekte - Hauptspeicher-Aufbau, Technologie von Speicher-Bauelementen - Rechner-Arithmetik, Mikroarchitektur und Instruktionssatz von Prozessoren - wichtige Komponenten, die den Programm-Ablauf wesentlich beeinflussen: Pipelining, Cache und Speicherverwaltung - Grundzüge des Betriebssystems, Prozesse, Scheduling, Synchronisation - Massenspeicher und Ein-Ausgabe-Einheiten 				
Teilnahmevoraussetzungen	keine				
Literaturangabe	div. Literatur über Rechneraufbau, Rechnerstrukturen u. Computer-Architektur				
Vergabe von Leistungspunkten	Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen (Prüfungsvorleistung) und Klausur (Prüfungsleistung). Die Modulnote ist die Note der Klausur				
Prüfungsformen und Leistungen	Modulabschlussprüfung: Klausur 180 Minuten				

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-06	Pflicht

Modultitel	Datenstrukturen und effiziente Algorithmen				
Empfohlen für	3. Semester				
Verantwortlich	Institut für Informatik				
Dauer	1 Semester				
Modulturnus	jedes Wintersemester				
Lehrformen		Titel	SWS	Präsenz- stunden	Selbst- studium
	Vorlesung	Datenstrukturen und effiziente Algorithmen	4	60	60
	Übung	Datenstrukturen und effiziente Algorithmen	2	30	60
	Praktikum	Datenstrukturen und effiziente Algorithmen	1	40	
Arbeitsaufwand	9 CP = 250 Arbeitsstunden (Workload)				
Verwendbarkeit	B.Sc. Informatik				
Ziele	Der Modul vermittelt die wichtigen Basisalgorithmen der Informatik. Das Grundwissen über effiziente Algorithmen und Datenstrukturen fördert die Problemlösungsfähigkeiten der Studierenden. Sie sollen in der Lage sein, einfache Probleme von der Auswahl der Verfahren bis zur effizienten Implementierung zu lösen. Für Lehramtsstudierende vermittelt das Modul somit Kenntnisse über grundlegende Problemstellungen der Informatik und dazugehörige Lösungsmöglichkeiten.				
Inhalt	<ul style="list-style-type: none"> • grundlegende Datenstrukturen und Algorithmen: Sortieren, Suchbäume, Prioritätswarteschlangen, Skiplisten, Hashing, Union-Find • Graphalgorithmen: Zusammenhangskomponenten, Wegesuche, Spannbäume, Matching, Flüsse • Paradigmen des Algorithmenentwurfs: Divide & Conquer, dynamisches Programmieren, randomisierte Algorithmen, Greedy Strategien • Analysetechniken: Analyse randomisierter Algorithmen, amortisierte Analyse 				
Teilnahmevoraussetzungen	Einführung in die Programmierung				
Literaturangabe	Algorithmen (Cormen, Leiserson, Rivest), Algorithmen in Java (Sedgewick), Algorithm Design (Kleinberg, Tardos)				
Vergabe von Leistungspunkten	Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen (Prüfungsvorleistung) und Klausur (Prüfungsleistung) Die Modulnote ist die Note der Klausur.				
Prüfungsformen und Leistungen	Modulabschlussprüfung: Klausur 180 Minuten				

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-10	Pflicht

Modultitel	Datenbanken				
Empfohlen für	4. Semester				
Verantwortlich	Institut für Informatik				
Dauer	2 Semester				
Modulturnus	jedes Sommersemester				
Lehrformen		Titel	SWS	Präsenz- stunden	Selbst- studium
	Vorlesung	Datenbanken I	2	30	40
	Übung	Datenbanken I	2	30	70
	Vorlesung	Datenbanken II	2	30	30
	Übung	Datenbanken II	1	15	45
	Praktikum	optional	2	80	10
Arbeitsaufwand	10 (13) CP = 290 (380) Arbeitsstunden (Workload)				
Verwendbarkeit	B.Sc. Informatik				
Ziele	<p>Teil 1: Datenbanktechnologie ist eine Schlüsseltechnologie der praktischen und angewandten Informatik. Datenbanken spielen in den Unternehmen eine immer zentralere Rolle, weil ein Großteil des „Wissens“ in Datenbanken gespeichert ist. Die Studierenden lernen den grundsätzlichen Aufbau von Datenbanken und ihre Benutzung kennen. Ebenso wird besonderer Wert auf die semantisch korrekte Modellierung eines Sachverhalts als Voraussetzung für den Datenbankentwurf gesehen. Einen weiteren Schwerpunkt bildet das Erlernen der Datenbanksprache SQL. Hierdurch sollen die Studierenden befähigt werden, die erworbenen Kenntnisse praktisch umzusetzen.</p> <p>Teil 2: Mit dieser Veranstaltung vertiefen die Studierenden ihre Kenntnisse im Datenbankbereich, insbesondere im Bereich Nicht-Standard-Datenbanken</p> <p>Praktikum: Im Rahmen des Praktikums sollen praktische Fertigkeiten bei der Erstellung und Verwendung relationaler Datenbanken erlernt werden. Darüberhinaus werden vertiefende Kenntnisse im Bereich kommerzieller Datenbanktechnologie erworben.</p>				
Inhalt	<p>Teil 1:</p> <ul style="list-style-type: none"> • Aufbau u. wesentliche Merkmale von Datenbankmanagementsystemen • Speichermodelle • DB-Modellierung nach dem Entity-Relationship-Modell u. UML-Modell • Relationale Algebra als Grundlage des Relationalen Modells • Relationenmodell u. Normalisierung des Datenmodells • SQL – als Datendefinitions-, Datenmanipulations- und Datenbankabfragesprache • Transaktionskonzept 				

	<ul style="list-style-type: none"> • Datenschutz u. Datensicherheit <p>Teil 2:</p> <ul style="list-style-type: none"> • Verteilte Datenbanken • Mobile Datenbanken • Temporale Datenbanken, Geodatenbanken • Objektorientierte Datenbanksysteme (OODBS): Grundlagen, Sprachen ODL, OQL) • Objektorientierte Datenbanksysteme / SQL99 & SQL 2003 • XML-Datenbanken: Speicherung von XML-Dokumenten, XML-Schema, X-Path, X-Query, XML-Datenbanksysteme • Datawarehouseing • Datamining <p>Praktikum: Anhand eines komplexen Beispiels erstellen die Studierenden einen Datenbankentwurf unter Verwendung der bekannten Techniken und führen die Implementierung in einem kommerziellen Datenbankmanagementsystem durch. Durch selbstgestelltes Skript ist die Datenbank mit Testdaten zu füllen und es ist eine Anwendungsschnittstelle zu entwickeln. Die einzelnen Phasen werden durch Präsentationen und Testate abgeschlossen. Die Aufgaben werden z.T. in größeren Gruppen bearbeitet (bis zu 15 Studierende), die dann die notwendigen Aufgaben in Kleingruppen weiter bearbeiten. Durch das Praktikum sollen die praktischen Fertigkeiten weiterentwickelt werden und die in der Vorlesung Datenbanken I vorgestellten Inhalte an einem praktischen Beispiel noch einmal im gegebenen Kontext umgesetzt werden.</p>
Teilnahmevoraussetzungen	Einführung in die Programmierung
Literaturangabe	<p>Vossen, G: Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme, Oldenbourg-Verlag Elmasri, Navathe: Grundlagewn von Datenbanksystemen, Pearson Studium Kemper,A, Eickler,A.: Datenbanksysteme, Oldenbourg-Verlag Martin,W: Data Warehousing, Data MIning – OLAP, Thomson Publishing Han, Kiawei, Kamber, Michelline: Data Mining, Morgan Kaufmann Publishers</p>
Vergabe von Leistungspunkten	<p>Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen (Prüfungsvorleistung) und Klausur (Prüfungsleistung) Die Modulnote ist die Note der Klausur.</p> <p>Praktikum: Erstellung vom Datenmodell, Umsetzung des Datenmodells unter Verwendung eines kommerziellen Datenbanksystems, Anwendungsentwicklung</p>
Prüfungsformen und Leistungen	<p>Praktikum: Die Prüfungsleistung wird nicht benotet. Präsentation, Ausarbeitung und Testate Modulabschlussprüfung: 2 Klausuren je 180 Minuten</p>

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-11	Wahlpflicht

Modultitel	Konzepte für Programmiersprachen				
Empfohlen für	Ab 4. Semester				
Verantwortlich	Prof. Dr. H. Göttler				
Dauer	1 Semester				
Modulturnus	Nach Bedarf				
Lehrformen		Titel	SWS	Präsenzstd.	Selbststudium
	Vorlesung	Konzepte für Programmiersprachen I	2	30	60
	Übung	Konzepte für Programmiersprachen I	2	30	60
	Vorlesung	Konzepte für Programmiersprachen II	2	30	60
	Übung	Konzepte für Programmiersprachen II	2	30	60
Arbeitsaufwand	12 CP = 360 Arbeitsstunden				
Verwendbarkeit	B.Sc. Informatik				
Ziele	<p>Teil 1: Kennenlernen der geschichtlichen Entwicklung der Höheren Programmiersprachen, des Aufbaus eines Compilers, der Methoden zur Beschreibung von Syntax und Semantik;</p> <p>Teil 2: Kennenlernen typischer Vertreter von Programmierparadigmen, die weder in „Programmiersprachen“ noch in „Einführung in die Programmierung“ behandelt werden; Kenntnisse über den Mechanismus von Laufzeitsystemen</p>				
Inhalt	<p>Teil 1: Entwicklung der Kontroll- und Datenstrukturen; Parameterübergabemechanismen; Phasen eines Compilers, Analysestrategien (top-down und bottom-up); operationelle, axiomatische und denotationale Semantik, Programmverifikation</p> <p>Teil 2: Algol-Familie, Cobol, Simula, Smalltalk, Fortran, Perl, Ada, C#, Programmiersprachen für verteilte Systeme, Laufzeitsysteme</p>				
Teilnahmevoraussetzungen	Einführung in die Programmierung, Programmiersprachen				
Literaturangabe	Lehrbücher wie Sebesta „Programming Languages“				
Vergabe von Leistungspunkten	Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen (Prüfungsvorleistung) und (bislang) mündliche Prüfung (Prüfungsleistung). Die Note der mündlichen Prüfung ist die Modulnote.				
Prüfungsformen und Leistungen	(bislang) mündliche Prüfung				

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-12	Wahlpflicht

Modultitel	Compilertechnik				
Empfohlen für	Ab 4. Semester				
Verantwortlich	Prof. Dr. H. Göttler				
Dauer	1 Semester				
Modulturnus	Jedes 2. – 3. Semester				
Lehrformen		Titel	SWS	Präsenzstd.	Selbststudium
	Vorlesung	Compilertechnik I	2	30	60
	Übung	Compilertechnik I	2	30	60
	Vorlesung	Compilertechnik II	2	30	60
	Übung	Compilertechnik II	2	30	60
Arbeitsaufwand	12 CP = 360 Arbeitsstunden				
Verwendbarkeit	B.Sc. Informatik				
Ziele	Teil 1: Funktionsweise eines Compilers, Einsatz von Generierungswerkzeugen, Transfer auf andere Gebiete				
Inhalt	Teil 1: Aufbau eines Compilers, Werkzeuge für die lexikalische Analyse wie Lex, Top-down-Verfahren (Tafelverfahren, Rekursiver Abstieg) für LL(k)-Grammatiken, attributierte Grammatiken Teil 2: LR(k)-Grammatiken, Einsatz von Werkzeugen wie Yacc, Laufzeitsysteme, Code-Generierung und --Optimierung				
Teilnahmevoraussetzungen	Einführung in die Programmierung, Programmiersprachen, Theoretische Grundlagen der Informatik I				
Literaturangabe	Lehrbücher wie Aho/Lam/Sethi/Ullman „Compilers“				
Vergabe von Leistungspunkten	Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen (Prüfungsvorleistung) und (bislang) mündliche Prüfung (Prüfungsleistung). Die Note der mündlichen Prüfung ist die Modulnote.				
Prüfungsformen und Leistungen	(bislang) mündliche Prüfung				

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-13	Wahlpflicht

Modultitel	Simulation				
Empfohlen für	5. Semester				
Verantwortlich	Prof. Dr. J. Perl				
Dauer	1 Semester				
Modulturnus	jedes Wintersemester				
Lehrformen		Titel	SWS	Präsenzstd.	Selbststudium
	Vorlesung mit integrierter Übung	Simulation	4	60	60
	Praktikum	Simulationswerkzeuge	2	40	60
	Seminar	Simulationstechniken in der Anwendung	2	30	80
Arbeitsaufwand	12 CP = 330 Arbeitsstunden (Workload)				
Verwendbarkeit	B. Sc. Informatik + Nebenfach				
Ziele	<p>Es sollen grundlegende Kenntnisse über Konzepte, Methoden und Werkzeuge der Simulation sowie entsprechende Einsatzmöglichkeiten durch Vertiefungen in typische Problemstellungen und deren methodisch-konzeptionelle Behandlung vermittelt werden.</p> <p>In den Übungen soll in Kleingruppen projekt- und teamorientiertes Arbeiten eingeübt werden.</p>				
Inhalt	<ul style="list-style-type: none"> - Analyse von Modellverhalten - Einführung in die Theorie der dynamischen Systeme - Einführung in Modell-Generatoren - Entwicklung und Einsatz von interaktiven Simulations-Modellen - Fallstudien und Beispielprojekte 				
Teilnahmevoraussetzungen	<p>Einführung in die Programmierung</p> <p>Mathematik für Informatiker</p> <p>Software Engineering</p> <p>Modellbildung</p>				
Literaturangabe	keine				
Vergabe von Leistungspunkten	<p>Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen (Prüfungsvorleistung) und Klausur (Prüfungsleistung)</p> <p>Die Modulnote ist die Note der Klausur.</p>				
Prüfungsformen und Leistungen	Modulabschlussprüfung: Klausur 180 Minuten				

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-14	Wahlpflicht

Modultitel	Modellbildung				
Empfohlen für	4. Semester				
Verantwortlich	Prof. Dr. J. Perl				
Dauer	1 Semester				
Modulturnus	jedes Sommersemester				
Lehrformen		Titel	SWS	Präsenzstd.	Selbststudium
	Vorlesung mit integrierter Übung	Modellbildung	4	60	100
	Seminar	Ausgewählte Kapitel der Modellbildung	2	30	70
Arbeitsaufwand	10 CP = 260 Arbeitsstunden (Workload)				
Verwendbarkeit	B. Sc. Informatik + Nebenfach				
Ziele	<p>Es sollen grundlegende Kenntnisse über Konzepte, Methoden und Werkzeuge der Modellbildung sowie entsprechende Einsatzmöglichkeiten durch Vertiefungen in typische Problemstellungen und deren methodisch-konzeptionelle Behandlung vermittelt werden.</p> <p>In den Übungen soll in Kleingruppen projekt- und teamorientiertes Arbeiten eingeübt werden.</p>				
Inhalt	<ul style="list-style-type: none"> - System- und Modellbegriff - Entwurf und Verifikation von Modellen - Entwicklung und Einsatz von Modellen - Grundlagen der computergestützten Simulation - Fallstudien 				
Teilnahmevoraussetzungen	Einführung in die Programmierung Mathematik für Informatiker Software Engineering				
Literaturangabe	keine				
Vergabe von Leistungspunkten	Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen (Prüfungsvorleistung) und Klausur (Prüfungsleistung) Die Modulnote ist die Note der Klausur.				
Prüfungsformen und Leistungen	Modulabschlussprüfung: Klausur 180 Minuten				

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-15	Wahlpflicht

Modultitel	Computergrafik				
Empfohlen für	4.-6. Semester				
Verantwortlich	Institut für Informatik				
Dauer	2 Semester				
Modulturnus	jedes Jahr				
Lehrformen		Titel	SWS	Präsenz- stunden	Selbst- studium
	Vorlesung	Computergrafik I	2	30	60
	Übung	Computergrafik I	2	30	60
	Vorlesung	Computergrafik II	2	30	60
	Übung	Computergrafik II	2	30	60
	Seminar (optional)	Computergrafik	2	30	60
	Praktikum (optional)	Computergrafik	2	30	60
Arbeitsaufwand	12 (+4+3) CP = 360 (+120) Arbeitsstunden (Workload)				
Verwendbarkeit	B.Sc. Informatik				
Ziele	Der Modul vermittelt die mathematischen Grundlagen zum Verständnis der Visualisierungstechniken in der Computergrafik. Die Studenten lernen anhand von einfachen Anwendungen, wie man zeitveränderliche, komplexe geometrische Szenen realistisch visualisiert oder mehrdimensionale wissenschaftliche Datensätze adäquat präsentiert.				
Inhalt	<ul style="list-style-type: none"> • affine und projektive Transformationen, elementare geometrische Algorithmen • Sichtbarkeitsberechnungen, Beleuchtungsmodelle, Texturen, Schatten • geometrisches Modellieren, parametrisierte Kurven und Flächen • Raytracing, Radiosity, Volumenvisualisierung • hardwareunterstützte Renderingtechniken in OpenGL/OpenSL • Animationstechniken, Kinematik und Dynamik von Starrkörpersystemen • Virtuelle Realität, Szenengraphen 				
Teilnahme- voraussetzungen	Einführung in die Programmierung, Mathematik für Informatiker I+II				
Literaturangabe	Grundlegende und aktuelle Originalarbeiten zu Themen der Computergrafik				
Vergabe von Leistungspunkten	Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen (Prüfungsvorleistung) und Klausur (Prüfungsleistung)				

	Die Modulnote ist die Note der Klausur.
Prüfungsformen und Leistungen	Modulabschlussprüfung: Klausur 120 Minuten

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-16	Wahlpflicht

Modultitel	Software-Architektur				
Empfohlen für	> 3. Semester				
Verantwortlich	Dr. Andreas Winter				
Dauer	1 Semester				
Modulturnus	Alternierend mit anderen Schwerpunktveranstaltungen der Softwaretechnik				
Lehrformen		Titel	SWS	Präsenzstd.	Selbststudium
	Vorlesung	Software-Architektur	2	30	45
	Übung	Software-Architektur	2	30	60
Arbeitsaufwand	6 CP = 165 Arbeitsstunden				
Verwendbarkeit	B.Sc. Informatik				
Ziele	<p>Software-Architekturen beschreiben die grundsätzliche Struktur komplexer Software-Systeme aus der Sicht verschiedener Stakeholder zu verschiedenen Zwecken. Sie werden jeweils abhängig vom Darstellungsziel durch ihre relevanten Komponenten, deren Beziehungen untereinander und deren Beziehungen zur Systemumwelt dargestellt.</p> <p>Diese Vorlesung behandelt die Darstellung, Entwicklung und Weiterentwicklung von Software-Architekturen. Sie soll den Hörern Kenntnisse vermitteln, Software-Architekturen zu erkennen, zu entwerfen und zu bewerten.</p>				
Inhalt	<ol style="list-style-type: none"> 1. Einführung <ol style="list-style-type: none"> 1.1. Bau-Architektur und Software-Architektur 1.2. Stakeholder (Auftraggeber, Entwickler, Benutzer) 1.3. Software-Architekt (Aufgaben und Fähigkeiten der Software-Architekten) 2. Software-Architektur in Prozessmodellen <ol style="list-style-type: none"> 2.1. Wasserfall-Modell 2.2. Rational Unified Process 2.3. eXtreme Programming 3. Entwurf von Software-Architekturen <ol style="list-style-type: none"> 3.1. Prinzipien für den Architektur-Entwurf 3.2. Qualitätsmerkmale für den Architektur-Entwurf 3.3. Aktivitäten für den Architektur-Entwurf 3.4. Heuristiken für den Architektur-Entwurf 4. Beschreibung von Software-Architekturen <ol style="list-style-type: none"> 4.1. Sichten und Sichtendefinitionen 4.2. Architektur-Strukturen 4.3. Architektur-Beschreibungsmittel (ADL, UML) 5. Architektur-Sichten <ol style="list-style-type: none"> 5.1. Innen und Außensicht 				

	<p>5.2. Siemens Sichten (Globale Analyse, Konzept-Sicht Modul-Sicht, Ausführungs-Sicht, Quelltext-Sicht)</p> <p>5.3. 4+1 Sichten</p> <p>6. Architektur-Muster</p> <p>6.1. Repository</p> <p>6.2. Layer</p> <p>6.3. Pipe-and-Filter</p> <p>6.4. Call-Return</p> <p>6.5. Manager</p> <p>6.6. Ereignis-gesteuerte Kontrolle</p> <p>7. Moderne Architekturen</p> <p>7.1. Komponenten-Architekturen</p> <p>7.2. Service-Orientierte Architekturen</p> <p>7.3. Produktlinien-Architekturen</p>
Teilnahme-voraussetzungen	Einführung in die Programmierung Einführung in die Softwareentwicklung
Literaturangabe	<p>Len Bass, Paul Clements, Ric Kazman: Software Architecture in Practice, Addison-Wesley, Boston, 2nd Edition, 2003.</p> <p>Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern-Oriented Software Architecture, A System of Patterns, Volume 1, John Wiley, 1996.</p> <p>Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Robert Nord, Judith Stafford: Documenting Software Architectures, Views and Beyond, Addison-Wesley, Boston, 2002.</p> <p>Christine Hofmeister, Robert Nord, Dilip Soni: Applied Software Architecture, Addison-Wesley, Reading, 1999.</p> <p>Ralf Reussner, Wilhelm Hasselbring: Handbuch der Software-Architektur, dPunkt, 2006.</p> <p>Chris Rupp, Jürgen Hahn, Stefan Queins, Mario Jeckle, Barbara Zengler: UML 2 glasklar. 2. Auflage, Hanser, 2005.</p>
Vergabe von Leistungspunkten	Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen und Vorlesung Die Modulnote ist die Note der Abschlussprüfung
Prüfungsformen und Leistungen	Mündliche (30 min) oder schriftliche (120 min) Abschlussprüfung

Bachelor of Science Informatik

Akademischer Grad	Modulnummer	Modulform
Bachelor of Science	M-17	Wahlpflicht

Modultitel	Kommunikationsnetze				
Empfohlen für	ab 3. Semester				
Verantwortlich	Institut für Informatik / Zentrum für Datenverarbeitung				
Dauer	1 Semester				
Modulturnus	jedes Sommersemester				
Lehrformen		Titel	SWS	Präsenzstd.	Selbststudium
	Vorlesung	Kommunikationsnetze	3	45	60
	Übung	Kommunikationsnetze	1	15	45
	Praktikum (optional)	Kommunikationsnetze	2	80	10
Arbeitsaufwand	6 (9)CP = 165 (255) Arbeitsstunden (Workload)				
Verwendbarkeit	B.Sc. Informatik, Diplom-Handelslehrer				
Ziele	Da die heutige IT ohne Rechner-Kommunikation nicht mehr denkbar ist, sollen die Studierenden ein Grundverständnis für die dabei stattfindenden Abläufe sowie den Aufbau, die Funktion und Administration von Datennetzen entwickeln. Dazu sollen neben den Konzepten und Verfahren der Datenkommunikation Kenntnisse der Protokolle, Dienste und Funktionsweise von Kommunikationsnetzen und ihrer Komponenten vermittelt werden.				
Inhalt	<ul style="list-style-type: none"> – Grundlagen der Datenkommunikation – OSI-Referenzmodell, TCP/IP-Modell – Netze/-Topologien – Protokolle und Dienste im Netz, – Netz-, Vermittlungsschicht/Routing, Transport-Protokolle – Höhere Dienste(ftp, http, smtp, ...) – Lokale /Weitverkehrsnetze: Komponenten, Typen (Ethernet,..., ATM, SDH) – Mobilkommunikation (WLAN, ...,Weitverkehrsnetze), ISDN,DSL 				
Teilnahmevoraussetzungen	keine				
Literaturangabe	div. Literatur zu Rechnernetzen				
Vergabe von Leistungspunkten	Leistungsnachweis nach erfolgreicher Teilnahme an den Übungen (Prüfungsvorleistung) und Klausur (Prüfungsleistung). Die Modulnote ist die Note der Klausur				
Prüfungsformen und Leistungen	Modulabschlussprüfung: Klausur 180 Minuten				