

Modulhandbuch

als Anlage zur
Prüfungsordnung
des

Bachelor-Studiengangs Informatik (Computer Science)

der Rheinisch-Westfälischen Technischen Hochschule Aachen

Stand: 1.Oktober 2006

Präambel

Das vorliegende Modulhandbuch für den Bachelor-Studiengang Informatik enthält die in der Bachelor-Prüfungsordnung aufgeführten Anlagen. Es kann durch Beschluss des Prüfungsausschusses Informatik ohne Änderung der Prüfungsordnung geändert werden. Die jeweils aktuelle Version des Modulhandbuchs ist über die Internetseiten der Bachelor-Studienberatung der Fachgruppe Informatik einsehbar.

Erläuterungen zu den Modulbeschreibungen

- Sofern nur ein Veranstalter angegeben ist, handelt es sich um den Modulverantwortlichen. Bei mehreren Dozenten ist der Modulverantwortliche durch Fettdruck gekennzeichnet. In den Fällen, in denen kein Modulverantwortlicher genannt ist (zum Beispiel beim Proseminar, das von allen Lehrenden parallel in eigener Verantwortung angeboten wird), ist der Prüfungsausschussvorsitzende verantwortlich für die Modulbeschreibung und Ansprechpartner für Fragen zum Modul.
- Die Lehrformen werden folgendermaßen abgekürzt: V = Vorlesung, Ü = Übung, P = Praktikum, S = Seminar. Die dazugehörige Semesterwochenstunden-Zahl (SWS) wird angehängt. V3+Ü2 bedeutet z.B. Vorlesungen im Umfang von 3 SWS und Übungen im Umfang von 2 SWS.

Inhalt

1.	Studienplan (Anlage 1 zur BPO).....	5
2.	Modulkatalog Pflichtveranstaltungen (Anlage 2 zur BPO)	6
2.1	Praktische Informatik	6
	Programmierung.....	6
	Datenstrukturen und Algorithmen.....	7
	Datenbanken und Informationssysteme	9
	Einführung in die Softwaretechnik	10
2.2	Technische Informatik.....	13
	Einführung in die Technische Informatik	13
	Elektrotechnische Grundlagen der Informatik	13
	Betriebssysteme und Systemsoftware	14
	Hardwarenahe Programmierung	15
	Sichere Verteilte Systeme	16
2.3	Theoretische Informatik	18
	Diskrete Strukturen	18
	Formale Systeme, Automaten, Prozesse (FoSAP).....	18
	Berechenbarkeit und Komplexität (BuK)	19
	Mathematische Logik.....	20
2.4	Mathematik.....	21
	Analysis für Informatiker	21
	Lineare Algebra.....	23
	Einführung in die angewandte Stochastik.....	24
	Numerisches Rechnen	25
2.5	Seminare, Praktika und Bachelorarbeit.....	26
	Einführung in das wissenschaftliche Arbeiten (Proseminar Informatik).....	26
	Software-Projektpraktikum	28
	Vertiefte Literatuarbeit und wissenschaftliche Präsentation (Seminar Informatik)	30
	Bachelor Project (Thesis).....	31
3.	Modulkatalog Wahlpflichtbereich (Anlage 3 zur BPO)	32
3.1	Theoretische Informatik	32
	Effiziente Algorithmen.....	32
	Einführung in den Compilerbau	33
	Introduction to Model Checking	34
	Einführung in die Funktionale Programmierung	35
	Einführung in die Logikprogrammierung	36

Angewandte Automatentheorie.....	37
3.2 Software und Kommunikation	39
Software-Architekturen.....	39
Distributed Applications and Middleware	39
Einführung in Eingebettete Software	41
3.3 Daten- und Informationsmanagement.....	42
Implementation of Databases	42
Introduction to Artificial Intelligence	43
Introduction to Knowledge Representation.....	44
Data Mining Algorithms	45
Web Engineering.....	46
3.4 Angewandte Informatik.....	47
Einführung in die Computergraphik	47
Designing Interactive Systems I.....	48
Introduction to High-Performance Computing	48
Computational Differentiation	49
4. Modulkatalog Anwendungsfächer (Anlage 4 zur BPO)	50
4.1 Betriebswirtschaftslehre	50
Grundzüge der Wirtschaftswissenschaften	50
Grundzüge des betrieblichen Rechnungswesens.....	51
Entscheidungslehre (Wirtschaftswissenschaften C).....	52
Quantitative Methoden (Operations Research)	53
4.2 Elektrotechnik	54
Grundgebiete der Elektrotechnik A.....	54
Grundgebiete der Elektrotechnik B	55
4.3 Mathematik.....	56
Algebra	57
Funktionentheorie I	57
Mathematisches Praktikum	58
Numerische Analysis I	58
Numerische Analysis II.....	59

1. Studienplan (Anlage 1 zur BPO)

	Praxis	Technik		Theorie	Mathematik		Studienleistungen		Wahl	Neben	Sum
WS 01	Prog (V4+Ü2) 8	TI (V2+Ü2) 4	ET* (V2+P2) 3	diStr. (V2+Ü1) 4	Ana (V4+Ü2) 8						27
SS 01	DSAL (V4+Ü2) 8			FSAP (V3+Ü2) 6	LA (V4+Ü2) 8	Num (V3+Ü2) 6	EinWA (ProS2) 2				33
WS 02		BS/SSW (V4+Ü2) 8	HW-PR (P2) 4	BuK (V3+Ü2) 6	Sto (V3+Ü1) 6					NF1 6	30
SS 02	DB (V3+Ü2) 6			Logik (V3+Ü2) 6			SWP (V1+P3) 7		WPF (V3+Ü2) 6	NF2 6	31
WS 03	SWT (V3+Ü2) 6			r-WPF (V3+Ü2) 6					2 WPF (V3+Ü2) 12	NF3 6	30
SS 03		SVS (V3+Ü2) 6					Thesis 12 Kolloquium 3	↑ Sem (S2) 4	↓	NF4 4	29
	28	28		28	28		28		18	22	180

Prog	Programmierung	diStr.	diskrete Strukturen
DSAL	Datenstrukturen und Algorithmen	FSAP	Formale Systeme, Automaten und Prozesse
DB	Datenbanken und Informationssysteme	BuK	Berechenbarkeit und Komplexität
SWT	Softwaretechnik	Logik	Logik
TI	Technische Informatik	Ana	Analysis
ET	Elektrotechnische Grundlagen inkl. Praktikum	LA	Lineare Algebra
BS/SSW	Betriebssysteme & Systemsoftware	Num	Numerisches Rechnen
HWP-PR	Praktikum: Hardware-nahes Programmieren	Sto	Stochastik
SVS	Sichere verteilte Systeme	EinWA	Einführung in das wissenschaftl. Arbeiten
WPF	Wahlpflichtfach aus Katalog	SWP	Software-Projekt
r-WPF	Wahlpflichtfach aus Theoriekatalog	NF	Nebenfach

2. Modulkatalog Pflichtveranstaltungen (Anlage 2 zur BPO)

2.1 Praktische Informatik

Titel	Programmierung
Cluster	Praktische Informatik
Credits	8 (240h, davon 90h Präsenz und 150h Eigenstudium)
Lehrformen/SWS	V4+Ü2
Sprache	Deutsch
Turnus	Jedes WS
Dozenten	Giesl , Bischof, Katoen, Lichter, Schroeder
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	---
Literatur	Folien und Skripte zur Vorlesung sowie z.B. folgende Bücher: <ul style="list-style-type: none"> • K. Echte & M. Goedicke: <i>Lehrbuch der Programmierung mit Java</i>, dpunkt Verlag, 2000 • R. Bird: <i>Introduction to Functional Programming Using Haskell</i>, Prentice Hall, 1998 • W. F. Clocksin & C. S. Mellish: <i>Programming in Prolog</i>, Springer, 2003
Lernziele	Erwerb der folgenden Kenntnisse und Fähigkeiten: <ul style="list-style-type: none"> • Beherrschung der wesentlichen Konzepte imperativer und objektorientierter Programmiersprachen sowie wichtiger Programmier Techniken in diesen Sprachen • Kenntnis der Programmierkonzepte logischer und funktionaler Programmiersprachen • Kenntnis grundlegender Datenstrukturen und ihrer Realisierung in verschiedenen Programmierparadigmen • Fähigkeit zur selbständigen Entwicklung kleinerer Programme und ihrer Dokumentation unter Beachtung üblicher Programmierkonventionen • Kenntnis grundlegender Beschreibungsformen für Programmiersprachen • Grundkenntnisse der Programmverifikation

Lerninhalte	<ul style="list-style-type: none"> - Sprachbeschreibung durch Grammatiken und Syntaxdiagramme - Imperative Programmierkonzepte: <ul style="list-style-type: none"> • Variablen- und Typkonzepte (z.B. primitive Datentypen, Arrays, Records, Enumerations, etc.) sowie Typkonversionen • Kontrollstrukturen (Sequenz, Verzweigung, Schleifen, etc.) • Grundlagen der Verifikation einfacher Programme • Pointer, Seiteneffekte und Grundlagen der Speicherverwaltung • Funktionen, Prozeduren und Parameterübergabeverfahren (call-by-value, call-by-reference) • Rekursive Funktionen und rekursive (lineare) Datenstrukturen (z.B. Listen, Stacks, Queues, etc.) • Grundlegende Beispielprogramme (z.B. einfache Such- und Sortieralgorithmen) - Objektorientierte Konzepte: <ul style="list-style-type: none"> • Vererbung, Polymorphie, Dynamisches Binden • Abstrakte Klassen und Interfaces • Grundlegende Programmiertechniken in imperativen und objektorientierten Sprachen (z.B. Datenabstraktion, Modularisierung, Schnittstellendokumentation, etc.) - Funktionale Konzepte: <ul style="list-style-type: none"> • Deklarationen, Ausdrücke, Pattern Matching, Auswertungsstrategien (call-by-value, call-by-name) • Typkonzepte und Polymorphie • Einfache Funktionen höherer Ordnung - Logische Konzepte: <ul style="list-style-type: none"> • Fakten und Regeln • Unifikation und Bearbeitung von Anfragen
Nachfolgemodule	<ul style="list-style-type: none"> • Datenstrukturen und Algorithmen • Software-Praktikum

Titel	Datenstrukturen und Algorithmen
Cluster	Praktische Informatik
Credits	8 (240h, davon 90h Präsenz und 150h Eigenstudium)
Lehrformen/SWS	V4+Ü2
Sprache	Deutsch
Turnus	Jedes SS
Veranstalter	Katoen, Kobbelt, Ney, Seidl, Rossmannith, Vöcking
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Beherrschung der wesentlichen Konzepte imperativer und objektorientierter Programmiersprachen sowie elementarer Programmiertechniken in diesen Sprachen (Vorlesung

	<p>Programmierung)</p> <ul style="list-style-type: none"> • Kenntnis linearer Datenstrukturen wie Arrays, Listen, Stacks, Queues (Vorlesung Programmierung) • Fähigkeit zur selbständigen Entwicklung kleinerer Programme (Vorlesung Programmierung) • Kenntnis elementarer diskreter Strukturen, insbesondere Graphen und Bäume (Vorlesung Diskrete Strukturen)
Literatur	<p>Folien und Skripte zur Vorlesung sowie z.B. folgende Bücher:</p> <ul style="list-style-type: none"> • T. Cormen, C. Leiserson, R. Rivest, C. Stein: <i>Introduction to Algorithms, 2nd Edition</i>, MIT Press and McGraw-Hill, 2001. • K. Mehlhorn and S. Näher: <i>The LEDA Platform of Combinatorial and Geometric Computing</i>, Cambridge University Press, 1999. • T. Ottmann, P. Widmayer: <i>Algorithmen und Datenstrukturen, 4. Auflage</i>, Spektrum Akademischer Verlag, 2002. • R Sedgewick: <i>Algorithms, 2nd Edition</i>, Addison-Wesley, 2002.
Lernziele	<p>Erwerb der folgenden Kenntnisse und Fähigkeiten:</p> <ul style="list-style-type: none"> • Kenntnis grundlegender Entwurfsmethoden für Algorithmen • Beherrschung einfacher und fortgeschrittener Methoden zur Laufzeitanalyse von Algorithmen • Verständnis der wesentlichen Komplexitätskategorien für Laufzeit und Speicherbedarf von Algorithmen • Kenntnis effizienter Algorithmen und Datenstrukturen für Standardprobleme • Fähigkeit der formalen Modellierung von algorithmischen Problemen sowie der Anpassung von vorhandenen Algorithmen und Datenstrukturen an die gegebene Problemstellung • Fähigkeit zur Implementierung der erlernten algorithmischen Methoden unter Berücksichtigung programmiertechnischer Konzepte wie z.B. die Kapselung von Datenstrukturen.
Lerninhalte	<p>Komplexität von Algorithmen</p> <ul style="list-style-type: none"> • Modelle für Laufzeit und Speicherplatz • Worst-Case- und Average-Case-Analysen • Asymptotische Komplexität („O-Notation“) • Komplexitätskategorien (z.B. exponentiell, polynomiell) <p>Allgemeine Entwurfs- und Analysemethoden</p> <ul style="list-style-type: none"> • Greedy-Algorithmen • Divide-and-Conquer-Verfahren • Dynamische Programmierung • Heuristische Ansätze (insbesondere Branch-and-Bound) • Lösen von Rekursiongleichungen (insbes. „Mastertheorem“) <p>Algorithmen für Sortierprobleme</p> <ul style="list-style-type: none"> • elementare Sortieralgorithmen (z.B. Insertionsort) • fortgeschrittene Sortierverfahren (Merge-, Quick-, Heapsort) • untere Schranke für vergleichsbasierte Sortierverfahren

	<ul style="list-style-type: none"> • Schlüsselbasiertes Sortieren (z.B. Bucketsort) • Order Statistics (z.B. Quickselect) <p>Datenstrukturen zur Verwaltung von Mengen</p> <ul style="list-style-type: none"> • Datenstrukturen für Mengen • Binäre Suchbäume • Balancierte Suchbäume • Priority Queues • Hashingverfahren <p>Graph- und Netzwerkalgorithmen</p> <ul style="list-style-type: none"> • Tiefensuche, Breitensuche • Bestimmung kürzester Wege • Berechnung minimaler Spannbäume • Einführung in Flussalgorithmen (Ford-Fulkerson-Methode) <p>Einführung in die algorithmische Geometrie, u.a.</p> <ul style="list-style-type: none"> • Sweepplintechnik • Bestimmung nächster Nachbarn
Nachfolgemodule	<p>Berechenbarkeit und Komplexität</p> <p>Datenbanken und Informationssysteme</p> <p>Effiziente Algorithmen (Wahlpflicht)</p>

Titel	Datenbanken und Informationssysteme
Cluster	Praktische Informatik
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	deutsch
Turnus	jedes WS
Veranstalter	Jarke, Seidl
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Datenstrukturen und Algorithmen • Grundlagen der Logik
Literatur	<p>Folien zur Vorlesung</p> <p>Standardbücher:</p> <ul style="list-style-type: none"> • Elmasri R., Navathe S.B., Fundamentals of Database Systems Benjamin-Cummings • Kemper, A., Eicker, A.: Datenbanksysteme – eine Einführung. Oldenbourg.

	<ul style="list-style-type: none"> • Vossen G., Datenmodelle, Datenbanksprachen und Datenbank-Managementsysteme, Addison-Wesley
Lernziele	<ul style="list-style-type: none"> • • Grundverständnis der Rolle von Datenbanken und Informationssystemen • • Gute Kenntnis und erste praktische Erfahrung mit dem relationalen Datenbankmodell, insbesondere den relationalen Anfragesprachen (SQL) und ihren formalen Grundlagen • • Grundkenntnisse der Vorgehensweise beim relationalen Datenbankentwurf, insbesondere konzeptuelle Modellierung und Normalisierungstheorie • • Verständnis der Grundprobleme und Ansätze der Datenbankimplementierung und Datenbankadministration (Architektur, Anfrageauswertung, Transaktionsmanagement) • • Grundüberblick über objektorientierte, objektrelationale und semi-strukturierte Datenmodelle sowie über Entwurf betrieblicher Informationssysteme • • Praktische Rechnererfahrung mit SQL, XML, ERP-Systemen
Lerninhalte	<ul style="list-style-type: none"> • Aufgaben und Bedeutung von Informationssystemen • Relationale Datenbankmodelle • Relationale Anfragesprachen und ihre formalen Grundlagen • Entwurf relationaler Datenbanken (konzeptuelle Modellierung, Normalisierungstheorie) • Grundelemente relationaler Datenbankimplementierung (Architekturen, Anfrageverarbeitung, Transaktionsmanagement) • Überblick neuere Datenmodelle: <ul style="list-style-type: none"> ○ objektorientierte / objektrelationale Datenbanken ○ Internet-Informationssysteme/ XML ○ Betriebliche Informationsmodellierung und ERP • Praktische Übungen im Datenbanklabor: SQL-Day, XML-Day, ERP-Day
Ggf. Nachfolgemodule	<p>Wahlvorlesungen zu Themen wie</p> <ul style="list-style-type: none"> • Implementierung von Datenbanken • Indexstrukturen, • Data Mining • Einführung Künstliche Intelligenz • Wissensrepräsentation • Betriebliche Informationssysteme <p>Bachelorarbeiten und Softwarepraktika im Bereich Datenbanken / Informationssysteme sowie Datenmanagement/ -exploration</p>

Titel	Einführung in die Softwaretechnik
Cluster	Kernbereich Praktische Informatik
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	Deutsch/Englisch
Turnus	Jedes WS

Veranstalter	Lichter, Nagl , Schroeder
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	Grundstudium der ersten 4 Semester
Literatur	<p>Bücher:</p> <ul style="list-style-type: none"> • H. Balzert: "Lehrbuch der Software-Technik 1", Spektrum Akadem. Verlag • C. Ghezzi, M. Jazayeri, D. Mandrioli: "Fundamentals of Software Engineering", Prentice Hall • Ludewig, H. Lichter: "Software-Engineering: Grundlagen, Mensch, Prozesse, Techniken“, dpunkt Verlag, 2006' • M. Nagl: "Softwaretechnik: Methodisches Programmieren im Großen", Springer, 1990, 2. Aufl. 2007 • I. Sommerville: "Software-Engineering", Addison-Wesley
Lernziele	<p>Lernziel der Vorlesung ist zum einen, den Softwareentwicklungs-Prozess sowie sein komplexes Produkt kennen zu lernen und zu charakterisieren. Zum anderen werden die Aktivitätenblöcke der Softwareentwicklung erörtert und Notationen für das Festhalten der Teilergebnisse sowie ihres Zusammenhangs eingeführt.</p> <p>Schließlich werden auch die Hauptklassen von Softwaresystemen skizziert.</p> <p>In den Übungen werden die angesprochenen Aspekte einzeln vertieft. Darüber hinaus ergeben die Resultate einiger Übungen ein größeres Beispiel. Schließlich tauchen Übungsaufgaben zu den Hauptklassen Transformationssysteme, Interaktive Systeme sowie eingebettete Systeme auf.</p>
Lerninhalte	<ul style="list-style-type: none"> • Einführung/Grundbegriffe Motivation, Realität, Einordnung, Vision • Aktivitäten und Dokumente im Software-Lebenszyklus Phasen, Arbeitsbereiche, Zusammenhang, Diskussion Lebenszyklus-Modelle • Der Entwicklungs- und Wartungsprozess: Allg. Aspekte Wartung, kritische Bereiche, Eigenschaften Programmsysteme, Modellierungsproblematik, Prinzipien der Modellierung, Prozesse/Konfigurationen, Statik/Dynamik • Requirements Engineering Klärung, Struktur des Prozesses, Gliederung Ergebnisse, Anforderungs-Spezifikation: Ermittlung, Perspektiven, Probleme, Rollen, Zusammenhang der Ergebnisse • Anforderungsspezifikation und Notationen Sprachen für das Requirements Engineering, Vorstellung einiger UML-Notationen, Probleme der Sprache/Methodik, kleine Fallstudie • Entwurf/Architekturerstellung

	<p>Software-Architekturen: Begriffsklärung, Bedeutung, Entwurfsprozess und Ergebnisse</p> <ul style="list-style-type: none"> • Notationen für Architekturen Sprachen für Architekturen, UML: Ergänzungen, Modulare Ansätze, Verteilung und techn. Architekturen • Formale Spezifikation Einordnung/Klassifikation, algebraische Spezifikation, Verhaltensspezifikation, operationale Spezifikation für Kernteile des Systems • Projektmanagement: Teilaspekte Gruppenmodelle, Aufwandsschätzverfahren, Konfigurationsverwaltung • Dokumentation Übersicht, Benutzerdokumentation, Entwicklungsdokumentation, neue Formen • Qualitätssicherung Klassifikation und häufigste Arten, Formen menschlicher Begutachtung, Allgemeines zu Test, Modul-/Teilsystem-, Integrations-, Abnahme-Test, Testplanung und Beendigung • Wartung Reverse-/Reengineering, Integration, Verteilung, Beispiele • Zusammenhang Meta-Modellierung, Modelltransformationen, MDA • Werkzeuge CASE-Tools, Entwicklungsumgebungen, Kritik des Stands
Ggf. Nachfolgemodule	<p>Software-Architekturen Einführung in eingebettete Systeme</p>

2.2 Technische Informatik

Titel	Einführung in die Technische Informatik
Cluster	Kernbereich Technische Informatik
Credits	4 (120h, davon 60h Präsenz und 60h Eigenstudium)
Lehrformen/SWS	V2 + Ü2
Sprache	Deutsch
Turnus	Jedes Wintersemester,
Veranstalter	Kowalewski , Lakemeyer, Spaniol
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	-
Literatur	<ul style="list-style-type: none"> • Oberschelp, Vossen: Rechneraufbau und Rechnerstrukturen. Oldenbourg-Verlag, München, 10. Aufl., 2006 • Becker, Drechsler, Molitor: Technische Informatik. Pearson, 2005.
Lernziele	Vermittlung grundlegender Kenntnisse über den Aufbau und die Funktionsweise von Digitalrechnern und ihrer Teile, sowie die mathematischen Hilfsmittel für ihre Beschreibung und ihren Entwurf.
Lerninhalte	<ul style="list-style-type: none"> • Schaltfunktionen und ihre Repräsentation • Spezifische Schaltnetze und ihre Verbesserung • Schaltnetzwerke • Rechnerarithmetik • Von-Neumann-Architektur, CISC/RISC
Nachfolgemodule	<ul style="list-style-type: none"> • Betriebssysteme und Systemsoftware (V+Ü) • Hardwarenahe Programmierung (P)

Titel	Elektrotechnische Grundlagen der Informatik
Cluster	Technische Informatik
Credits	6 (180h, davon 60h Präsenz und 120h Eigenstudium)
Lehrformen/SWS	V2 (Wintersemester) +P2 (Sommersemester)
Sprache	Deutsch
Turnus	Jährlich
Veranstalter	Kowalewski
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	-
Literatur	R. Paul: Elektrotechnik und Elektronik für Informatiker. Teubner, 1994.

Lernziele	<p>Erwerb elementarer Kenntnisse über die physikalischen Prinzipien, die der Funktionsweise von elektronischen Rechnern zugrunde liegen, über die wichtigsten Technologien und über die wichtigsten Konzepte, die beim Entwurf und der Analyse von rechnergestützten Systemen benötigt werden. Zweck ist neben grundlegendem Verständnis auch die Befähigung, mit Ingenieuren kompetent kommunizieren zu können.</p> <p>Der Vorlesungsteil dient dazu, die in dem Praktikum geübten Themen einzuführen.</p>
Lerninhalte	<ul style="list-style-type: none"> • Auffrischung Physik-Grundwissen (Ladung, Feld, Potenzial, Spannung, Strom, Widerstand, Ohmsches Gesetz, Spannungsteiler, Kirchhoffsche Regeln, Kapazität, Kondensator, Ladekurve, RC-Tiefpass, Induktivität, RLC-Schwingkreis) • Halbleiter-Bauelemente (pn-Übergang, Diode, Kennlinie, Anwendungen: Gleichrichter, UND/ODER-Schaltungen, Bipolartransistor, Kennlinie, physikalische Erklärung (nnp, pnp), Anwendungen: Schalter, Flipflop) • Speichertechnologien (RAM, ROM, EPROM, EEPROM, FLASH) • Programmierbare Logik (PAL, PLA, PLD, CPLD, FPGA) • Hardwareentwurf (Einführung in VHDL, Synthese eines einfachen Schaltwerkes (z.B. Automat, Zähler, digitales Filter) in VHDL) • Analoge Schaltungen (Motivation: Anbindung des Rechners an seine Umgebung; Operationsverstärker, Grundsaltungen: Komparator, Schmitt-Trigger, Analog-Digital- und Digital-Analogwandlung mit Operationsverstärkern, pulsweitenmodulierte Signale) • Mikrocontroller (Architektur, Programmierung, Anwendungen)
Nachfolgemodule	<ul style="list-style-type: none"> • Betriebssysteme und Systemsoftware • Hardwarenahe Programmierung

Titel	Betriebssysteme und Systemsoftware
Cluster	Kernbereich Praktische Informatik
Credits	8 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V4+Ü2
Sprache	Englisch
Turnus	Jedes WS
Veranstalter	Spaniol , Kowalewski
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Vorlesung Rechnerstrukturen

Literatur	<p>Folien zur Vorlesung</p> <p>Empfohlene Bücher:</p> <ul style="list-style-type: none"> • Spaniol, O.; Popien, C.; Reichl, P.; Schuba, M.: Systemprogrammierung. Skript zur Vorlesung, Aachener Beiträge zur Informatik, Band 14. • Silberschatz, A.; Galvin, P.; Gagne, G.: Operating System Concepts with C & C++. 7th Edition, John Wiley & Sons, 2004. • Tanenbaum, A.S.: Modern Operating Systems. Prentice-Hall, 2001.
Lernziele	<p>Erwerb der folgenden Kenntnisse und Fähigkeiten:</p> <ul style="list-style-type: none"> • Kenntnisse grundlegender Konzepte des Zusammenwirkens der Bestandteile eines Rechners • Kenntnisse des Zusammenspiels zwischen Hardware und Software • Kenntnisse effizienter Ressourcenverwaltung • Fähigkeit zur effizienten Entwicklung komplexer Systeme
Lerninhalte	<ul style="list-style-type: none"> • Aufgaben und Struktur von Betriebssystemen • Das Betriebssystem Unix • Prozesse und Nebenläufigkeit • Synchronisation und Kommunikation • CPU-Scheduling • Speicherverwaltung • Dateisysteme und Dateiverwaltung • Rechteverwaltung und Zugriffskontrolle • Systemaufrufe, Shells, Utilities • Assemblerprogrammierung • Prozeduraufrufe • Stack- und Heapverwaltung, Garbage-Collection • E/A-System • Überblick: Compiler-Binder-Lader
Ggf. Nachfolgemodule	<ul style="list-style-type: none"> • Praktikum Systemprogrammierung • Sichere Verteilte Systeme

Titel	Hardwarenahe Programmierung
Cluster	Technische Informatik
Credits	4 (120h, davon 30h Präsenz und 90h Eigenstudium)
Lehrformen/SWS	P2
Sprache	Deutsch
Turnus	Jedes Wintersemester,
Veranstalter	Kowalewski

Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Einführung in die Technische Informatik (Rechnerstrukturen) (V+Ü) Hardware-Grundlagen (V+P)
Literatur	Skript
Lernziele	Praktische Erprobung des Stoffes aus <i>Elektrotechnische Grundlagen</i> und des Teils über digitale Grundschaltungen aus <i>Rechnerstrukturen</i> . Der Ort im Curriculum, wo Studierende, die noch nie ein Multimeter, ein Oszilloskop oder einen integrierten Schaltkreis in der Hand hatten, die Gelegenheit dazu bekommen. Sie sollen lernen, wie man systematisch Versuche durchführt (d.h. vorbereitet, ausführt und auswertet). Vermittlung von „Social Skills“ durch Gruppenarbeit.
Lerninhalte	Assemblerprogrammierung eines Mikroprozessors (entweder kommerzielles Board oder selbst entworfener Prozessor auf PLD aus vorherigem Praktikum), z.B. Einlesen von Messwerten, Verarbeiten (Filtern o.ä.) und Anzeigen auf Display oder Steuerreaktion (Anschalten eines Ventilators bei Überschreitung eines Temperaturgrenzwertes o.ä.) <ul style="list-style-type: none"> • Erstes Programm: Maschinencode, von Hand geladen • Realisierung eines einfachen Linkers/Laders • Einfacher Scheduler • Ein-/Ausgabeeanbindung Interrupts/Polling • ggfls. einfache Speicherverwaltung • Anbindung des Prozessors an Peripheriebausteine z.B. Anschluss an A/D-Wandler zur Messwernerfassung und-verarbeitung, Ansteuerung von Leistungsausgängen über D/A-Wandler
Nachfolgemodule	<ul style="list-style-type: none"> • Betriebssysteme und Systemsoftware (V+Ü)

Titel	Sichere Verteilte Systeme
Cluster	Kernbereich Praktische Informatik
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	Englisch
Turnus	Jedes WS
Veranstalter	Spaniol
Leistungen	Klausurarbeit oder mündliche Prüfung

Vorwissen	Grundstudium der ersten 4 Semester
Literatur	<p>Folien zur Vorlesung</p> <p>Empfohlene Bücher:</p> <ul style="list-style-type: none"> • A. S. Tanenbaum: <i>Computer Networks</i>, 4. Auflage, Prentice-Hall International, 2002 • Jochen Schiller: <i>Mobile Communications</i>, 2. Auflage, Addison Wesley, 2004 • Kaufman, C., Perlman, R., Speciner, M., <i>Network Security, Private Communication in a Public World</i>, Prentice Hall, Englewood Cliffs, 1995
Lernziele	<p>Erwerb folgender Kenntnisse und Fähigkeiten:</p> <ul style="list-style-type: none"> • Kenntnis des Aufbaus von Kommunikationsprotokollen • Kenntnis der Protokolle lokaler Netze und Weitverkehrsnetze • Kenntnis der Geräte zum Aufbau lokaler Netze • Kenntnis gängiger Internet-Protokolle • Kenntnis der Sicherheitsprobleme gängiger Protokolle sowie möglicher Angriffsszenarien • Fähigkeit zum Aufbau eines lokalen Netzes • Fähigkeit zur Programmierung kommunizierender Anwendungen basierend auf Internet-Protokollen
Lerninhalte	<ul style="list-style-type: none"> • OSI-Referenzmodell und TCP/IP-Referenzmodell • Client/Server- und Peer-to-Peer-Systeme • Sicherheitsmanagement und Datenschutz • Netzkomponenten und Firewalls • Lokale Netze, speziell Ethernet und Wireless LAN • Weitverkehrsnetze, speziell ATM und SDH • Internet-Protokolle: IP, TCP/UDP und Anwendungsprotokolle, Sicherheitsprobleme und Angriffe • Grundlagen der Kryptographie und sichere Protokolle • Quality of Service im Internet • Grundlagen des Netzmanagements
Ggf. Nachfolgemodule	<p>Da in dieser Veranstaltung nur Grundkenntnisse vermittelt werden können, kann eine Vertiefung in mehreren Spezialvorlesungen erfolgen:</p> <ul style="list-style-type: none"> • Multimedia Systems • Security in Communication Networks • Modelling and Evaluation of Communication Networks • Distributed Applications and Middleware

2.3 Theoretische Informatik

Titel	Diskrete Strukturen
Cluster	Grundstudium (1. Semester), Theoretische Informatik
Credits	4 (120h, davon 45h Präsenz und 75h Eigenstudium)
Lehrformen/SWS	V2+Ü1
Sprache	Deutsch
Turnus	Jedes WS
Veranstalter	Triesch , weitere Dozenten der Mathematik
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	
Literatur	<ul style="list-style-type: none"> • A. Steger, Diskrete Strukturen, Bd.1, Springer 2001
Lernziele	Beherrschung elementarer mathematischer Begriffsbildungen im Kontext informatischer Anwendungen
Lerninhalte	<ul style="list-style-type: none"> • Mengen, Funktionen, Relationen anhand informatischer Beispiele • Boolesche Algebra • Endliche Kombinatorik • Elementare Zahlentheorie

Titel	Formale Systeme, Automaten, Prozesse (FoSAP)
Cluster	Grundstudium (2. Semester), Theoretische Informatik
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	Deutsch
Turnus	Jedes SS
Veranstalter	Thomas , Katoen, Giesl, Vöcking, Rossmann
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	Elementare mathematische Begriffe (aus 1. Semester)
Literatur	Skript und Folien zur Vorlesung Standardbücher: <ul style="list-style-type: none"> • Hopcroft, Motwani, Ullman, Introduction to Automata, Theory, Languages, and Computation, Addison-Wesley 2001 (Ch.1-7) • M. Sipser, Introduction to the Theory of Computation, PWS Publ. Comp. 1997, Part 1.

Lernziele	<p>Beherrschung elementarer Darstellungs- und Modellierungstechniken der Informatik, angebunden an konkrete Beispiele.</p> <ul style="list-style-type: none"> • Syntaxdefinitionen durch Regelsysteme und ihre Anwendung • Automaten als Grundstruktur zustandsbasierter Systeme • Einfache Modelle der Nebenläufigkeit (synchronisierte Produkte, Petrinetze) • Kenntnis der fundamentalen Algorithmen dazu (Transformation und Analyseverfahren für Automaten und Regelsysteme)
Lerninhalte	<p>I. Formale Systeme: Terme, Wörter, Sprachen anhand von Kernbeispielen: u.a. Zahlterme, arithmetische und boolesche Terme, while-Programme. Definition von Termmengen und Programmiersprachen durch Regelsysteme (Termersetzungssysteme, Grammatiken), Ableitungsbegriff, Methode der strukturellen Induktion. Klassifikation von Grammatiken (Chomsky-Hierarchie) und elementare Sachverhalte zu kontextfreien Grammatiken: Normalformen, Wortproblem (Ableitbarkeitstest), Nichtleerheitstest.</p> <p>II. Automaten: Endliche Automaten (deterministisch, nichtdeterministisch), Abschlusseigenschaften (u.a. Produktautomaten), reguläre Ausdrücke, Nichtleerheits- und Äquivalenztest, Nachweis nichtregulärer Sprachen. Kellerautomaten (deterministisch und nichtdeterministisch), Übersetzung von kontextfreien Grammatiken in Kellerautomaten als Beispiel der Implementierung von Rekursion durch Kellerspeicher.</p> <p>III. Prozesse: Elementare Modellierungsformen verteilter und nebenläufiger Systeme: Synchronisierte Produkte, Petrinetze und kommunizierende sequentielle Prozesse (CSP). Vorstellung und Einübung anhand von Beispielen, Vergleich mit dem Grundmodell des endlichen Automaten.</p>
Ggf. Nachfolgemodule	Vorlesungen „Berechenbarkeit und Komplexität“ (BuK) und „Logik“

Titel	Berechenbarkeit und Komplexität (BuK)
Cluster	Grundstudium (3. Semester), Theoretische Informatik
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	Deutsch
Turnus	Jedes WS
Veranstalter	Vöcking , Rossmannith, Thomas, Katoen, Giesl
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	Vorlesungen „Diskrete Strukturen“, „Formale Systeme, Automaten, Prozesse“ (aus 1. und 2. Semester)
Literatur	Skript und Folien zur Vorlesung

	Standardbücher: <ul style="list-style-type: none"> • J. Hromkovic, Theoretische Informatik, 2. Aufl., Teubner-Verlag 2004 • I. Wegener, Theoretische Informatik, 2. Aufl., Teubner-Verlag 1999
Lernziele	<ul style="list-style-type: none"> • Präzisierung und Tragweite des Algorithmusbegriffs • Begriffsbildungen zur prinzipiellen Lösbarkeit algorithmischer Probleme • Grundlagen zur Berechnungskomplexität • Approximation als Ansatz zur Lösung schwerer Probleme
Lerninhalte	<ul style="list-style-type: none"> • Beispiele algorithmischer Probleme, Darstellung durch Sprachen und Funktionen, Frage der Lösbarkeit • Turingmaschinen, Church-Turing-These • Berechenbarkeit, Entscheidbarkeit, Aufzählbarkeit • Simulationen zwischen verschiedenen Berechnungsmodellen, universelle Maschinen bzw. Programme • Unentscheidbare Probleme (u.a. Postisches Korrespondenz-Problem) • Komplexitätsklassen und elementare Sachverhalte zu Zeit- und Platzkomplexität • Polynomielle Reduktionen und NP-Vollständigkeit • Approximation als Methode zur Lösung NP-harter Probleme, Beispiel eines Polynomzeit-Approximationsschemas (FPTAS)
Ggf. Nachfolgemodule	Mathematische Logik

Titel	Mathematische Logik
Cluster	Grundstudium (4. Semester), Theoretische Informatik
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
SWS	V3 + Ü2
Sprache	Deutsch
Turnus	Jedes SS
Veranstalter	E. Grädel
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	Mathematische Grundkenntnisse, insbesondere aus den Vorlesungen Diskrete Strukturen und Lineare Algebra (aus 1. und 2. Semester), Grundkenntnisse über Berechenbarkeit und Komplexität.
Literatur	Skript zur Vorlesung Als zusätzliches Lehrbuch wird speziell empfohlen:

	H.-D. Ebbinghaus, J. Flum, W. Thomas, Einführung in die mathematische Logik, 4.Aufl. Spektrum Akad. Verlag 1996.
Lernziele	<ul style="list-style-type: none"> • Sachverhalte in geeigneten logischen Systemen zu formalisieren und mit diesen Formalisierungen umzugehen; • grundlegende Begriffe und Methoden der mathematischen Logik zu verstehen (Syntax und Semantik logischer Systeme, Folgerungsbeziehung, Erfüllbarkeit, Beweiskalküle, Definierbarkeit, etc.); • die Ausdrucksstärke und Grenzen logischer Systeme beurteilen zu können; • einige der fundamentalen Resultate der mathematischen Logik des 20. Jahrhunderts (z.B. Vollständigkeitssatz, Kompaktheitssatz, Unentscheidbarkeit der Prädikatenlogik) kennenzulernen und ihre Bedeutung für Mathematik und Informatik zu verstehen.
Lerninhalte	<ul style="list-style-type: none"> • Aussagenlogik (Grundlagen, algorithmische Fragen, Kompaktheit, Resolution, Sequenzenkalkül) • Strukturen, Syntax und Semantik der Prädikatenlogik • Einführung in weitere Logiken (modale und temporale Logiken, Logiken höherer Stufe) • Auswertungsspiele, Modellvergleichsspiele • Beweiskalküle, Termstrukturen, Vollständigkeitssatz • Kompaktheitssatz und Anwendungen • Entscheidbarkeit, Unentscheidbarkeit und Komplexität von logischen Spezifikationen

2.4 Mathematik

Modulname:	Analysis für Informatiker
Studiengang:	Informatik
Verantwortliche:	Prof. Dr. Dr.h.c. Hubertus Th. Jongen Prof. Dr. Aloys Krieg
weitere Dozenten:	Prof. Dr. Hennig Esser Prof. Dr. Rudolf Stens

Lernziele:	<p>Die Studierenden sollen:</p> <p>Verständnis für die Grundlagen der Analysis, wie Grenzwert, Stetigkeit, Differentiation und Integration entwickeln. die Fähigkeit zum selbständigen Umgang mit den Inhalten der Lehrveranstaltung erwerben und lernen die grundlegenden Techniken der Analysis sicher zu beherrschen. Intuition für die mathematische Denkweise entwickeln und deren Umsetzung in präzise Begriffe und Begründungen einüben. exemplarisch die Entwicklung der Analysis an einigen zentralen Begriffen nachvollziehen, z.B. Iterationsverfahren und Rückkopplung bei der Lösung von nichtlinearen Gleichungen. durch Klausurtraining ein Gespür für den Umfang und Schwierigkeitsgrad einer schriftlichen Klausur sowie eine Einsicht in mögliche Lösungsdarstellung bekommen. das Basiswissen und die wesentlichen Fertigkeiten aus dem Bereich der Analysis für das weitere Studium erwerben.</p>
Modulinhalte:	<p>1. Reelle Zahlen 2. Folgen reeller Zahlen 3. Reelle Funktionen 4. Differentiation 5. Integration (Riemannsches Integral) 6. Reihen reeller Zahlen 7. Folgen und Reihen von Funktionen 8. Uneigentliche Integrale 9. Funktionen mehrerer Veränderlicher</p>
Einordnung:	Grundmodul, Bachelor 1. oder 2. Semester
Modulvoraussetzungen:	keine
Lehrformen:	Vorlesung (4 SWS), Übung (2 SWS), Klausurtraining
Sprache:	deutsch
Literatur:	H. Esser und H. Th. Jongen, <i>Analysis für Informatiker</i> , Skript
Leistungsnachweise:	Schriftliche Klausur (120 Min.)
ECTSPunkte (Aufwand):	8 (240h, davon 90h Präsenz und 150h Eigenstudium)
Dauer des Moduls:	Ein Semester
Häufigkeit des Angebots:	Jedes Wintersemester

Titel	Lineare Algebra
Cluster	Mathematik
Credits	8 (240h, davon 90h Präsenz und 150h Eigenstudium)
Lehrformen/SWS	V4+Ü2
Sprache	Deutsch
Turnus	Jedes SS
Veranstalter	Dozenten der Mathematik
Leistungen	Klausurarbeit
Vorwissen	<ul style="list-style-type: none"> • Analysis • Diskrete Strukturen
Literatur	<ul style="list-style-type: none"> • G. Fischer, Lineare Algebra, Vieweg, 2000 • K. Jänich, Lineare Algebra, Springer, 2001 • S. Lang, Lineare Algebra, 3rd Ed., Springer, 1989 • F. Lorenz, Lineare Algebra I, Spektrum, 1992 • C. Meyer, Matrix Analysis and Applied Linear Algebra, Wellesley-Cambridge Press, 2003 • L. Trefethen, D. Bau, Numerical Linear Algebra, SIAM, 1997
Lernziele	<ul style="list-style-type: none"> • Verständnis für lineare Zusammenhänge • Ausprägung von mathematischer Intuition und geometrischer Vorstellungskraft • Kenntnis von algebraischen Strukturen am Beispiel • Einblick in die Anwendungen der linearen Algebra durch Vorstellung ausgewählter Problemstellungen • Erkennen des Bezugs zu numerischen Verfahren
Lerninhalte	<ul style="list-style-type: none"> • Körper und Polynomring • Vektorräume, lineare Abbildungen und Matrizen • Basis, Dimension und Rang • Lineare Gleichungssysteme <ul style="list-style-type: none"> ○ Lösungsmengen, über- und unterbestimmte Systeme ○ Gauß-Algorithmus und LU-Zerlegung ○ Inverse und Pseudoinverse • Determinanten • Eigenwerte und Eigenvektoren, Diagonalisierung • Bilinearformen und quadratische Formen, Skalarprodukte, Orthogonalität • Gram-Schmidt Verfahren, QR-Zerlegung, Singulärwertzerlegung • Spektralsatz (Hauptachsentransformation) • Diskrete Fouriertransformation

Nachfolgemodul	Numerisches Rechnen
----------------	---------------------

Titel	Einführung in die angewandte Stochastik
Cluster	Mathem. Grundlagen
Credits	6 (180h, davon 60h Präsenz und 120h Eigenstudium)
Lehrformen/SWS	V3+Ü1
Sprache	Deutsch
Turnus	Jedes SS
Veranstalter	Kamps, Bock, Cramer
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	Kenntnisse aus Mengenlehre und Analysis (Reihen, Riemann-Integration)
Literatur	Skript zur VL, Standardbücher?
Lernziele	<ul style="list-style-type: none"> • Intuition für die statistische Denkweise entwickeln und deren Umsetzung in präzise Begriffe und Begründungen einüben. • Exemplarisch die Entwicklung der Wahrscheinlichkeitsrechnung und Statistik an einigen Anwendungen nachvollziehen. • Die Fähigkeiten zum selbständigen Umgang mit den Inhalten der Lehrveranstaltung erwerben und lernen die grundlegenden Techniken der Wahrscheinlichkeitsrechnung und Statistik sicher zu beherrschen. • Das Basiswissen und die wesentlichen Fertigkeiten aus dem Bereich der Wahrscheinlichkeitsrechnung und Statistik für das weitere Studium erwerben.
Lerninhalte	<ol style="list-style-type: none"> 1. Einleitung 2. Wahrscheinlichkeitsrechnung <ol style="list-style-type: none"> a. Wahrscheinlichkeitsräume <ol style="list-style-type: none"> i. Grundlagen der Wahrscheinlichkeitsrechnung: (Mengentheoretische Grundlagen, Kolmogorov-Axiome, Laplace-Modell, Grundformeln der Kombinatorik) ii. Diskrete Wahrscheinlichkeitsmaße: (Binomialverteilung, Poisson-Verteilung, Geometrische Verteilung, ...) iii. Eigenschaften von Wahrscheinlichkeitsmaßen iv. Bedingte Wahrscheinlichkeiten v. Stochastische Unabhängigkeit von Ereignissen vi. Wahrscheinlichkeitsmaße mit Riemann-Dichten: Exponential-, Weibull-, Gamma-, Normal-Rechteckverteilung, ...

	<ul style="list-style-type: none"> b. Zufallsvariablen <ul style="list-style-type: none"> i. Zufallsvariablen und Wahrscheinlichkeitsmaße ii. Verteilungsdichte, Verteilungsfunktion und Quantilfunktion iii. Mehrdimensionale Zufallsvariablen: gemeinsame Verteilung mehrdimensionale Normalverteilung, Randverteilung bedingte Verteilung, Produkträume iv. Transformation von Zufallsvariablen: (Dichtetransformationssatz, Faltung) v. Erwartungswerte, Varianz, Kovarianz und Korrelation vi. Erzeugende Funktionen und Laplace-Transformation vii. Bedingte Erwartungswerte <p>3. Statistik</p> <ul style="list-style-type: none"> a. Grundlegende Methoden der Beschreibenden Statistik <ul style="list-style-type: none"> i. Einführung und Grundbegriffe ii. Lage- und Streuungsmaße iii. Empirische Verteilungsfunktion iv. Klassierte Daten und Histogramm v. Zusammenhangsmaße vi. Regressionsanalyse b. Elementare Verfahren der Schließenden Statistik <ul style="list-style-type: none"> i. Problemstellungen der schließenden Statistik ii. Parameterschätzungen: Erwartungstreue, Güte und Konsistenz iii. Schätzung der Verteilungsfunktion iv. Maximum-Likelihood-Schätzung v. Konfidenzintervalle vi. Schätzungen bei Normalverteilung vii. Zentraler Grenzwertsatz viii. Lineare Regressionsmodelle ix. Elemente der Bayes-Statistik: Bayessche Entscheidungstheorie, Parameter- und Bereichsschätzung, Schätzung einer Wahrscheinlichkeit
Ggf. Nachfolgemodule	

Titel	Numerisches Rechnen
Cluster	Mathematik
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	Deutsch
Turnus	Jedes SS, Wiederholungsprüfung im WS
Veranstalter	Dozenten der Mathematik

Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Analysis • Lineare Algebra
Literatur	<ul style="list-style-type: none"> • Lloyd Trefethen and David Bau: Numerical Linear Algebra, 1997, SIAM • Charles Van Loan: Introduction to Scientific Computing, 2000, Prentice Hall • Differentialgleichungen und Numerik für Informatiker und Physiker, H. Esser und H. Th. Jongen, Skript
Lernziele	<ul style="list-style-type: none"> • Kenntnis der wichtigsten Standardverfahren zur Lösung numerischer Kernprobleme • Verständnis zur Auswahl eines effizienten Verfahrens für eine gegebene Problemstellung • Praktische Erfahrung beim Lösen einfacher Modellprobleme mittels Matlab.
Lerninhalte	<ul style="list-style-type: none"> • Numerische Lineare Algebra • Matrixfaktorisierungen • Nichtlineare Gleichungssysteme • Interpolation und Integration von Funktionen • Gewöhnliche Differentialgleichungen • Partielle Differentialgleichungen • Optimierung (Newton-Verfahren, Levenberg-Marquardt) • Numerische Software
Mögliche Nachfolgemodule	Introduction to High-Performance Computing, Combinatorial Problems in Scientific Computing, Computational Differentiation, Computergrafik, Data Mining, Sprachverarbeitung, Mustererkennung

2.5 Seminare, Praktika und Bachelorarbeit

Modul	Einführung in das wissenschaftliche Arbeiten (Proseminar Informatik)
Cluster	Schlüsselqualifikationen (Soft Skills, Social Skills)
Studiengang, Semester	Bachelor Informatik, 2. oder 3. Semester
Credits	2 (60h, davon 30h Präsenz und 30h Eigenstudium)
Lehrformen/SWS	S2

Sprache	Deutsch
Turnus	Jedes Semester (WS + SS)
Veranstalter	Alle Dozenten der Fachgruppe Informatik
Leistungen	Schriftliche Ausarbeitung (1.000 – 2.500 Wörter) sowie mündliche Präsentation (15-30 Minuten), aktive Teilnahme an Diskussionen, aktive Teilnahme an der Schulung „Literaturrecherche für Informatiker“ in der Informatikbibliothek.
Vorwissen	Grundkenntnisse der Informatik aus Modulen des 1. oder 2. Semesters (abhängig vom konkret angebotenen Thema)
Literatur	Themenabhängig; wird vorgegeben bzw. selbst recherchiert
Lernziele	<p>Erwerb der folgenden Kenntnisse und Fähigkeiten, um Konzepte und Methoden der Informatik wissenschaftlich darzustellen:</p> <ul style="list-style-type: none"> • Kenntnis von Methoden zur Literaturrecherche in physischen und elektronischen wissenschaftlichen Bibliotheken sowie die Fähigkeit, sich eigenständig in ein vorgegebenes Thema der Informatik durch Auswahl und Aufbereitung geeigneter Literatur einzuarbeiten (Schulung und Recherche anhand individuell abgestimmter Recherchebeispiele durch die Informatikbibliothek). • Fähigkeit, ein vorgegebenes Thema der Informatik anschaulich und mit angemessenen Formalismen termingerecht und in definiertem Umfang schriftlich auszuarbeiten; Beachtung korrekter Zitierungstechniken; Nachweis der eigenständigen Erarbeitung durch Darstellung selbst gewählter geeigneter Beispiele. • Fähigkeit, die anschauliche mündliche Präsentation eines Themas der Informatik unter Einsatz geeigneter Medien und Beispiele in vorgegebener Dauer zu planen und durchzuführen. • Fähigkeit, sich aktiv an Diskussionen über Themen der Informatik in Präsenzveranstaltungen zu beteiligen. • Ggf. Fähigkeit, eine Gruppenentscheidung zur Abgrenzung und Aufteilung eines Themas für mehrere Bearbeiter in abgeschlossene Teilthemen herbeizuführen.
Lerninhalte	Das Erreichen der Lernziele wird durch Einübung an Hand persönlich zugeordneter Themen der Informatik sowie die aktive Teilnahme an den Präsentationsterminen verfolgt. Die Wahl der Themengebiete obliegt dem jeweiligen Veranstalter.

Titel	Software-Projektpraktikum
Cluster	Sonstige
Credits	7 (210h, davon 60h Präsenz und 150h Eigenstudium)
Lehrformen/SWS	P3+V1
Sprache	Deutsch/Englisch
Turnus	Jedes SS
Veranstalter	Dozenten der Fachgruppe Informatik
Leistungen	Lösen einer mittelschweren Softwareaufgabe im Team, Präsentation der Ergebnisse, insbesondere Absprachen bzgl. technischer Schnittstellen, aber auch bzgl. Arbeitsorganisation
Vorwissen	Inhalte der ersten 3 Semester, insbesondere Programmierung, Datenstruktur und Algorithmen sowie Hardwarenahe Programmierung
Literatur	Wird jeweils bekannt gegeben.
Veranstaltungsform und -organisation	<p>Das Projektpraktikum findet an der RWTH dezentral an den Lehrstühlen/Lehreinheiten statt. Die Dozenten bieten unterschiedliche Projektpraktika an, deren Anzahl insgesamt so bemessen ist, dass die Studierenden eines Jahrgangs versorgt werden können. Typischerweise hat ein Projektpraktikum 25 Teilnehmer, die sich auf etwa 3 Gruppen verteilen. Diese Gruppen bearbeiten parallel und in Konkurrenz die gestellte Software-Problematik. Innerhalb der Gruppen wird die jeweilige Lösung arbeitsteilig erarbeitet.</p> <p>Die Lösung des Software-Projektpraktikums wird somit in Gruppen von etwa 8 Studierenden erarbeitet. Jeder der Teilnehmer einer Gruppe führt im Praktikum mind. zwei Rollen (Anforderungsspezifikator, Architekt, Implementierer, Tester usw.) aus. Die Rollenzuteilung übernehmen die Teilnehmer selbst. Dadurch ist das Kennenlernen der im Folgenden erwähnten technischen als auch sozialen Schnittstellenprobleme garantiert.</p> <p>Es ergeben sich verschiedene Praktika, je nach Ausrichter (z.B. eingebettetes System, paralleles System, objektorientiertes System, Datenbanksystem, Werkzeug usw.). Auch die verwendeten Ansätze, Methoden, Programmiersprachen und Entwicklungs-Werkzeuge sind i.a. unterschiedlich. Deshalb findet zu Beginn des Praktikums eine Veranstaltung in Form einer Vorlesung statt, in der die Aufgabenzergliederung, die erwarteten Dokumentstrukturen sowie die spezifischen Kenntnisse vermittelt werden. Es werden dabei insbesondere auch einfache Regeln des Projektmanagements (Aufgabenverteilung) vermittelt.</p> <p>Der Inhalt dieser Veranstaltungen ist z.T. allgemeingültig, z.T. aber auch spezifisch. Die gemeinsamen Teile werden einmalig ausgearbeitet und allen Praktikumsteilnehmern vermittelt. Für die spezifischen Teile wird es eine Verabredung geben, welche Aspekte in dieser Veranstaltung zu erörtern sind. Beides sichert - trotz der Verschiedenartigkeit der Praktikumsthemen - den nötigen Standard der Vorbereitung.</p>

Lernziele	<p>Der Schwerpunkt des Software-Projekts liegt zum einen darin, den Teilnehmern fundierte Entwicklungskennntnisse zu vermitteln. Dies geschieht dadurch, dass ein größeres Programmsystem, das aus mehreren Bestandteilen besteht, erstellt wird. Die Teilnehmer arbeiten sich dazu in die verwendete Programmiersprache – sofern diese neu ist – ein und sie wissen, wie diese anzuwenden ist. Weiterhin erlernen sie den Umgang mit modernen Entwicklungswerkzeugen, die Dokumentation sowie die Präsentation der erarbeiteten Ergebnisse. Um die Ergebnisse systematisch zu prüfen, führen die Teilnehmer Software-Inspektionen und -Tests durch. Kurzum: Die Teilnehmer lernen, dass eine Lösung nur zu erzielen ist, wenn Schnittstellen sorgfältig geplant, formuliert und eingehalten werden.</p> <p>Die Teilnehmer lernen insbesondere die mit der Arbeitsteiligkeit verbundenen gruppensdynamischen Effekte kennen (Ergebnis trifft nicht oder verspätet ein, auf das gewartet werden muss, Teilnehmer muss zur Lieferung „animiert“ werden etc.). Das Eintreten dieser Effekte ist insoweit garantiert, als jede Gruppe die Arbeitsteiligkeit selbst managen soll. Neben den gruppensdynamischen Problemen werden Abstimmungen und Präsentationen eingeübt. Die Vorstellung von Ergebnissen erfolgt in der Gruppe, aber auch im Plenum. Dies verbessert Vortrags- und Präsentationstechnik.</p>
Lerninhalte	<ul style="list-style-type: none"> • Umgangssprachliche Formulierung der Anforderungen • Fundierte Kenntnisse in einer Programmiersprache • Entwurf einfacher Software-Architekturen • Implementierung gemäß Programmierrichtlinien • Entwicklung und Durchführung von Software-Tests • Prüfung der erarbeiteten Ergebnisse durch Inspektionen • Systematische, strukturierte Dokumentation des Codes sowie der vorausgehenden Anforderungen bzw. Architektur • Umgang mit einer modernen Entwicklungsumgebung • Präsentation der erarbeiteten Ergebnisse • Gruppensdynamische Effekte bei arbeitsteiliger Bearbeitung
Ggf. Nachfolgemodule	Einführung in die Softwaretechnik sowie weitere Veranstaltungen zu den Themen, für die das Projektpraktikum durchgeführt wurde.

Modul	Vertiefte Literaturarbeit und wissenschaftliche Präsentation (Seminar Informatik)
Cluster	Schlüsselqualifikationen (Soft Skills, Social Skills)
Studiengang, Semester	Bachelor Informatik, 5.-6. Semester
Credits	4 (davon 2 für Schlüsselqualifikationen)
Lehrformen/SWS	S2
Sprache	Deutsch oder Englisch
Turnus	Jedes Semester (WS + SS)
Veranstalter	Alle Dozenten der Fachgruppe Informatik
Leistungen	Schriftliche Ausarbeitung (2.500 – 7.500 Wörter) sowie mündliche Präsentation (45-60 Minuten), aktive Teilnahme an Diskussionen
Vorwissen	Abhängig vom konkret angebotenen Themengebiet werden unterschiedliche Vorkenntnisse aus Moduln vorausgesetzt, die vom jeweiligen Dozenten vorab festgelegt und bekanntgegeben werden.
Literatur	Themenabhängig; wird vorgegeben bzw. selbst recherchiert
Lernziele	<p>Erwerb der folgenden Kenntnisse und Fähigkeiten, um Konzepte, Vorgehensweisen und Ergebnisse eines wissenschaftlichen Themas der Informatik aufzubereiten und zu präsentieren:</p> <ul style="list-style-type: none"> • Fähigkeit, sich auf der Basis geeigneter Literatur, insbesondere wissenschaftlicher Originalartikel, eigenständig in ein fortgeschrittenes Thema der Informatik einzuarbeiten, das Thema geeignet einzuordnen und einzugrenzen sowie eine kritische Bewertung zu entwickeln. • Fähigkeit, die Konzepte, Vorgehensweisen und Ergebnisse eines vorgegebenen Themas der Informatik anschaulich und mit angemessenen Formalismen termingerecht und in definiertem Umfang vertieft schriftlich auszuarbeiten; Nachweis der eigenständigen Erarbeitung durch Darstellung selbst gewählter Beispiele. • Fähigkeit, die anschauliche mündliche Präsentation eines vertieften Themas der Informatik unter Einsatz geeigneter Medien und Beispiele in vorgegebener Dauer zu planen und durchzuführen. • Fähigkeit, sich aktiv an Diskussionen zu vertieften Themen der Informatik in Präsenzveranstaltungen zu beteiligen.
Lerninhalte	Das Erreichen der Lernziele wird durch Einübung an Hand persönlich zugeordneter vertiefter wissenschaftlicher Themen sowie die aktive Teilnahme an den Präsentationsterminen verfolgt. Die Wahl der Themengebiete obliegt dem jeweiligen Veranstalter.
Nachfolgemodule	

Modul	Bachelor Project (Thesis)
Studiengang, Semester	Bachelor Informatik, 6. Semester
Credits	15 (davon 3 für Schlüsselqualifikationen)
Lehrformen/SWS	
Sprache	Deutsch oder Englisch
Turnus	Jedes Sommersemester; Ausweichmöglichkeit im Wintersemester
Veranstalter	Alle Dozenten der Fachgruppe Informatik
Leistungen	Schriftliche Ausarbeitung (20-60 Seiten) sowie mündliche Präsentation (20-40 Minuten); aktive Teilnahme an Präsentations- und Diskussionsveranstaltungen
Voraussetzungen	Zum Bachelor Project wird zugelassen, wer mindestens 120 ECTS aus den Modulen der vorhergehenden Semester erreicht hat. Für konkrete Aufgabenstellungen werden unterschiedliche Vorkenntnisse benötigt, die vom jeweiligen Betreuer festgelegt werden.
Literatur	Themenabhängig; wird vorgegeben bzw. selbst recherchiert
Lernziele	Erwerb folgender Fähigkeiten, um ein wissenschaftsnahes Thema der Informatik eigenständig auszuarbeiten: <ul style="list-style-type: none"> - Fähigkeit, sich in das Thema einzuarbeiten, es einzuordnen, einzugrenzen, kritisch zu bewerten und weiter zu entwickeln. - Fähigkeit, das Thema anschaulich und formal angemessen in einem bestimmten Umfang schriftlich darzustellen. - Fähigkeit, das Thema fachgerecht und anschaulich in einem Vortrag einer bestimmten Dauer zu präsentieren. - Fähigkeit, aktiv zu Diskussionen über wissenschaftsnahen Themen der Informatik beizutragen.
Lerninhalte	Für das Bachelor-Projekt wird ein wissenschaftsnahes Thema zu Konzepten, Vorgehensweisen und Ergebnissen der Informatik mit dem Betreuer vereinbart. Das Thema kann theoretisch oder praktisch orientiert sein, in jedem Fall ist eine kritische Auseinandersetzung und Bewertung gefordert. Beispiele sind etwa: <ul style="list-style-type: none"> - Literaturüberblick und Bewertung bestehender Ansätze zu einem aktuellen wissenschaftlichen Themengebiet; vertiefte Bewertung und analytischer oder empirischer Vergleich von ausgewählten Lösungskonzepten. - Implementierung, Weiterentwicklung und Evaluierung von bestehenden Verfahren und Konzepten der Informatik zur wissenschaftlichen Analyse (Evaluierungsprototyp) oder zur didaktischen Verwendung (Demonstrationsprototyp); Evaluierung der Leistungsfähigkeit von Systemen in Bezug auf bestimmte Aufgabenstellungen und Arbeitslasten. Themengebiete können gemeinschaftlich bearbeitet werden, müssen jedoch in Absprache mit dem Betreuer in individuell vertieften und abgegrenzten Leistungen resultieren.

3. Modulkatalog Wahlpflichtbereich (Anlage 3 zur BPO)

3.1 Theoretische Informatik

Modul	Effiziente Algorithmen
Cluster	Wahlpflicht Informatik / Effiziente Algorithmen
Studiengang- und Semester	Bachelor Informatik 4.-6. Semester
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3 Ü2
Sprache	Deutsch
Turnus	Jedes SS
Veranstalter	Prof. Rossmannith, Prof. Vöcking
Leistungen	Klausurarbeit oder mündliche Prüfung
Voraussetzungen	Kenntnisse aus den Modulen <i>Algorithmen und Datenstrukturen</i> sowie <i>Berechenbarkeit und Komplexität</i>
Literatur	Zur Vorlesung wird ein Skript erstellt und folgende Literatur empfohlen: <ul style="list-style-type: none"> • T. Cormen, C. Leiserson, R. Rivest, C. Stein: <i>Introduction to Algorithms, 2nd Edition</i>, MIT Press and McGraw-Hill, 2001. • C. Papadimitriou and K. Steiglitz: <i>Combinatorial Optimization: Algorithms and Complexity</i>, Dover Publications, Inc., 1998. • V. Vazirani, <i>Approximation Algorithms</i>, Springer, 2001. • R. Motwani, P. Raghavan. <i>Randomized Algorithms</i>, Cambridge University Press, 1995
Lernziele	<ul style="list-style-type: none"> • Kenntnis und Beherrschung der fortgeschrittener Methoden aus der Algorithmik und kombinatorischen Optimierung • Fähigkeit Probleme geeignet zu modellieren und rigorose Lösungen im mathematischen Modell zu erarbeiten

Lerninhalte	<p>Algorithmen für Flussprobleme</p> <p>Algorithmen für Zuordnungsprobleme (Matchings)</p> <p>Lineare Programmierung</p> <ul style="list-style-type: none"> • Simplexverfahren • Ellipsoidmethode • Dualitätsprinzip • Aspekte der Ganzzahligkeit <p>Methoden und Techniken für schwierige Probleme</p> <ul style="list-style-type: none"> • Approximationsalgorithmen (u.a. LP-basierte und primal-duale Verfahren) • Parametrisierte Algorithmen • Universelle heuristische Methoden <p>Randomisierte Algorithmen</p> <p>Online Algorithmen</p>
Nachfolgemodule	<ul style="list-style-type: none"> • Bachelor Project

Titel	Einführung in den Compilerbau
Cluster	Wahlpflicht Theoretische Informatik
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	Deutsch/Englisch
Turnus	Jährlich
Veranstalter	Katoen, Naumann, Noll
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Beherrschung der wesentlichen Konzepte imperativer und objektorientierter Programmiersprachen sowie elementarer Programmieretechniken in diesen Sprachen (Modul Programmierung) • Kenntnis von Datenstrukturen wie Listen, Stacks, Queues und Bäumen (Modul Datenstrukturen und Algorithmen) • Kenntnis grundlegender Automatenmodelle wie endliche Automaten und Kellerautomaten (Modul Formale Systeme, Automaten und Prozesse)
Literatur	<p>Folien und Skripte zur Vorlesung sowie folgende Lehrbücher:</p> <ul style="list-style-type: none"> • A. Aho, R. Sethi, J. Ullman: <i>Compilers – Principles, Techniques, and Tools</i>. Addison-Wesley, 1988. • A.W. Appel, J. Palsberg: <i>Modern Compiler Implementation in Java</i>. Cambridge University Press, 2002. • D. Grune, H.E. Bal, C.J.H. Jacobs, K.G. Langendoen: <i>Modern Compiler Design</i>. Wiley & Sons, 2000. • R. Wilhelm, D. Maurer: <i>Übersetzerbau</i>, 2. Auflage. Springer, 1997.

Lernziele	<p>Erwerb der folgenden Kenntnisse und Fähigkeiten:</p> <ul style="list-style-type: none"> • Verständnis der Konstruktion und Wirkungsweise von Compilern für höhere Programmiersprachen • Kenntnisse über Methoden der Syntaxbeschreibung (reguläre Ausdrücke, kontextfreie und attributierte Grammatiken, EBNF) • Fähigkeit zur Implementierung einfacher Compilerkomponenten (Scanner, Parser) • Kenntnisse im Einsatz compilererzeugender Werkzeuge
Lerninhalte	<ul style="list-style-type: none"> • Lexikalische Analyse von Programmen (Scanner) • Syntaktische Analyse von Programmen (Parser) • Semantische Analyse • Werkzeuge zur Compilerkonstruktion (lex, yacc)

Titel	Introduction to Model Checking
Cluster	Wahlpflicht Theoretische Informatik
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	Deutsch/English
Turnus	Jährlich
Veranstalter	Katoen , Thomas
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Kenntnis grundlegender Automatenmodelle wie endliche Automaten und Kellerautomaten (Modul Formale Systeme, Automaten und Prozesse) • Kenntnis der Aussagenlogik (Modul Mathematische Logik) • Kenntnis von Datenstrukturen wie Stacks, Bäumen und Graphen und deren elementarer Algorithmen (Modul Datenstrukturen und Algorithmen)
Literatur	<p>Folien und Skripte zur Vorlesung sowie folgende Lehrbücher:</p> <ul style="list-style-type: none"> • C. Baier, J.-P. Katoen: <i>Principles of Model Checking</i>, 2006. • M. Huth and M.D. Ryan: <i>Logic in Computer Science, Modelling and Reasoning about Systems</i>, Cambridge Univ. Press, 2004. • E.M. Clarke, O. Grumberg, D. Peled: <i>Model Checking</i>, MIT Press, 1999.
Lernziele	<p>Erwerb der folgenden Kenntnisse und Fähigkeiten:</p> <ul style="list-style-type: none"> • Modellierung von (nebenläufigen) Programmen • Kenntnisse über Eigenschaftsklassen • Verständnis der Konstruktion und Wirkungsweise von Model-checking Algorithmen für LTL und CTL • Fähigkeit zum Einsatz eines Model Checkers (Spin)

Lerninhalte	Folgende Hauptthemen:
Nachfolgemodule	<ul style="list-style-type: none"> • Transitionsysteme • Eigenschaftsklassen: Safety, Liveness, Invarianten und Fairness • Linear Temporal Logic (LTL) • Computational Tree Logic (CTL) • Model-checking Algorithmen für LTL und CTL

Titel	Einführung in die Funktionale Programmierung
Cluster	Theoretische Informatik
Studiengang	Bachelor Informatik (Wahlpflicht)
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	Deutsch oder Englisch
Turnus	ca. alle 2 Jahre
Veranstalter	Giesl
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Beherrschung der wesentlichen Konzepte der Programmierung (Vorlesung Programmierung) • Erste Grundkenntnisse in einer funktionalen Programmiersprache sind hilfreich, aber nicht notwendig (Vorlesung Programmierung)
Literatur	<p>Skript und Folien zur Vorlesung sowie z.B. folgende Bücher:</p> <ul style="list-style-type: none"> • R. Bird: <i>Introduction to Functional Programming Using Haskell</i>, Prentice Hall, 1998. • P. Pepper: <i>Funktionale Programmierung</i>, Springer, 2002. • C. Reade: <i>Elements of Functional Programming</i>, Addison-Wesley, 1989. • P. Thiemann: <i>Grundlagen der Funktionalen Programmierung</i>, Teubner, 1994.

Lernziele	<p>Erwerb der folgenden Kenntnisse und Fähigkeiten:</p> <ul style="list-style-type: none"> • Kenntnis der Programmier Techniken in funktionalen Programmiersprachen • Kenntnis der Konzepte, die funktionalen Programmiersprachen zu Grunde liegen • Fähigkeit zur formalen Festlegung der Semantik funktionaler Programmiersprachen • Fähigkeit zur Implementierung funktionaler Sprachen
Lerninhalte	<p>Einführung in die Programmiersprache Haskell</p> <ul style="list-style-type: none"> • Syntax der verschiedenen Sprachkonstrukte • Funktionen höherer Ordnung • Programmieren mit Lazy Evaluation <p>Denotationelle Semantik funktionaler Programme</p> <ul style="list-style-type: none"> • Vollständige Ordnungen und Fixpunkte • Denotationelle Semantik von Haskell <p>Der Lambda-Kalkül</p> <ul style="list-style-type: none"> • Syntax und operationelle Semantik des Lambda-Kalküls • Reduzierung von Haskell auf den Lambda-Kalkül
Nachfolgemodule	

Titel	Einführung in die Logikprogrammierung
Cluster	Theoretische Informatik
Studiengang	Bachelor Informatik (Wahlpflicht)
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	Deutsch oder Englisch
Turnus	ca. alle 2 Jahre
Veranstalter	Giesl
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Beherrschung der wesentlichen Konzepte der Programmierung (Vorlesung Programmierung) • Erste Grundkenntnisse in einer logischen Programmiersprache sind hilfreich, aber nicht notwendig (Vorlesung Programmierung) • Erste Grundkenntnisse der Prädikatenlogik sind hilfreich, aber nicht notwendig (Vorlesung Mathematische Logik)

Literatur	<p>Skript und Folien zur Vorlesung sowie z.B. folgende Bücher:</p> <ul style="list-style-type: none"> • I. Bratko: <i>Prolog Programming for Artificial Intelligence</i>, Addison-Wesley, 2001. • W. F. Clocksin, C. S. Mellish: <i>Programming in Prolog</i>, Springer, 2003. • M. Hanus: <i>Problemlösen mit Prolog</i>, Teubner, 1987. • J. W. Lloyd: <i>Foundations of Logic Programming</i>, Springer, 1993. • P. H. Schmitt: <i>Theorie der logischen Programmierung</i>, Springer, 1992.
Lernziele	<p>Erwerb der folgenden Kenntnisse und Fähigkeiten:</p> <ul style="list-style-type: none"> • Kenntnis der Programmier Techniken in logischen Programmiersprachen • Kenntnis der Konzepte und der prädikatenlogischen Grundlagen logischer Programmiersprachen • Fähigkeit zur Implementierung logischer Sprachen • Fähigkeiten zum Einsatz logischer Programmiersprachen in verschiedenen Anwendungsbereichen
Lerninhalte	<p>Prädikatenlogische Grundlagen</p> <ul style="list-style-type: none"> • Unifikation • Resolution • Horn-Klauseln und SLD-Resolution <p>Logikprogramme</p> <ul style="list-style-type: none"> • Auswertungsstrategien <p>Die Programmiersprache Prolog</p> <ul style="list-style-type: none"> • Negation as Failure • Nicht-logische Bestandteile von Prolog • Programmier Techniken <p>Anwendungen und Erweiterungen der Logikprogrammierung</p>
Nachfolgemodule	

Titel	Angewandte Automatentheorie
Cluster	Wahlpflicht Theoretische Informatik
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
SWS	V3+Ü2
Sprache	Deutsch
Turnus	2-Jährlich
Veranstalter	Thomas

Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	Pflichtvorlesungen des Clusters „Theoretische Informatik“
Literatur	<ul style="list-style-type: none"> • W. Thomas, Applied Automata Theory, RWTH Aachen 2004 • J. Berstel, Transductions and Context-Free Languages, Teubner 1979 • A. Paz, Introduction to Probabilistic Automata, Acad. Press 1971 • H. Straubing, Finite Automata, Formal Logic, and Circuit Complexity, Birkhäuser 1994
Lernziele	<p>Methodenwissen für Umgang mit</p> <ul style="list-style-type: none"> • verallgemeinerten Automatenmodellen (gewichtete Automaten, Transducer), • alternativen Beschreibungsformen (reguläre Ausdrücke, Schaltkreise, Logiken) • fundamentalen Algorithmen (Minimierung, Äquivalenztest, exemplarische Lernverfahren) <p>Illustration der Konzepte und Verfahren anhand von Beispielen aus allen Bereichen der Informatik.</p>
Lerninhalte	<ul style="list-style-type: none"> • Gewichtete Automaten (einschließlich probabilistischer Automaten), ihre Verhaltensbeschreibung und elementare Algorithmen zur Verhaltensanalyse. • Transduktionen, ihre Klassifizierung und ihre Anwendung z.B. in der Text- und Sprachverarbeitung. • Spezielle Klassen regulärer Sprachen und Automaten (Zusammenhang mit Programmkomplexität, Pattern Matching, Schaltkreiskomplexität, Logik-Beschreibungen) • Alternierende Automaten • Simulation, Bisimulation und die effiziente Minimierung von Automaten • Algorithmisches Lernen im Kontext der Automatentheorie
Nachfolgemodule	<ul style="list-style-type: none"> • Seminar über Automatentheorie

3.2 Software und Kommunikation

Titel	Software-Architekturen
Cluster	Software und Kommunikation
Studiengang und Semester	Bachelor 6. Semester
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	Deutsch/Englisch
Turnus	Sommersemester jedes 2. Jahres
Veranstalter	Nagl, Lehrstuhl Informatik 3 (Softwaretechnik)
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	Grundstudium Bachelor; Einführung in die Softwaretechnik
Literatur	M. Nagl: Methodisches Programmieren im Großen, Springer-Verlag, 1990, 2. Aufl. 2007 weitere schriftliche Unterlagen andere Lehrbücher zur Ergänzung
Lernziele	Objektbasierte und objektorientierte Architekturmodellierung Integrierter Ansatz aus Lokalität, Schichtung, Vererbung Kennenlernen großer Beispiele für Transformationssysteme, interaktive Systeme sowie eingebetteter Systeme Anwendbarer Ansatz auch für Reverse Engineering Integrationsfragestellungen, eingebetteter Systeme
Lerninhalte	<ul style="list-style-type: none"> • Modellieren auf Entwurfsebene • ein Modulkonzept • Teilarchitekturüberlegungen • Übertragung in Programmiersprachen • einige Architekturbeispiele
Nachfolgemodule	Bachelor Thesis

Titel	Distributed Applications and Middleware
-------	--

Cluster	PI
Credits	6 (180h, davon 60h Präsenz und 120h Eigenstudium)
Lehrformen/SWS	V3+Ü1
Sprache	Englisch
Turnus	Jedes zweite SS
Veranstalter	Spaniol
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Grundkenntnisse in Rechnernetzen und Kommunikationsprotokollen (Vorlesung Sichere Verteilte Systeme) • Grundkenntnisse zu Betriebssystemen (Vorlesung Betriebssysteme und Systemsoftware) • Fähigkeit zur selbständigen Entwicklung kleinerer Programme (Vorlesung Programmierung)
Literatur	<p>Folien zur Vorlesung sowie als (freiwillige) Ergänzung folgende Bücher:</p> <ul style="list-style-type: none"> • A.S. Tanenbaum, M. van Steen: <i>Distributed Systems - Principles and Paradigms</i>. Prentice-Hall, 2002 • G. Coulouris, J. Dollimore, T. Kindberg: <i>Distributed Systems - Concepts and Design</i>. Addison-Wesley, 2001 • Z. Tari, O. Bukhres: <i>Fundamentals of Distributed Object Systems - The CORBA Perspective</i>. Wiley, 2001
Lernziele	<p>Erwerb der folgenden Kenntnisse und Fähigkeiten:</p> <ul style="list-style-type: none"> • Kenntnis grundlegender Konzepte zur Kommunikation in verteilten Systemen • Kenntnis von Mechanismen zum dynamischen Binden verteilter Objekte (Name-, Directory- und Discovery-Services) • Kenntnis von Algorithmen zur Synchronisation, Koordination und Replikation verteilter Objekte • Kenntnis gängiger Middleware-Technologien • Fähigkeit zur Auswahl von geeigneten Synchronisations- und Koordinationsalgorithmen zu gegebenen Problemsituationen • Fähigkeit zur Entwicklung verteilter Anwendungen basierend auf den vermittelten Middleware-Technologien
Lerninhalte	<p>Kommunikation in verteilten Systemen</p> <ul style="list-style-type: none"> • Das Client/Server-Modell • Remote Procedure Call (RPC) und Remote Method Invocation (RMI) • Nachrichten-basierte Systeme <p>Namensdienste und ähnliche Konzepte</p> <ul style="list-style-type: none"> • Grundlegende Mechanismen von Namensdiensten • Das Domain Name System • Directory Services am Beispiel von X.500 und LDAP • Discovery Services am Beispiel von Jini

	<ul style="list-style-type: none"> • Lokalisierungsdienste <p>Uhrensynchronisation in verteilten Systemen</p> <ul style="list-style-type: none"> • Synchronisation mit einer Referenzuhr: Cristian's Algorithmus, der Berkeley-Algorithmus und das Network Time Protocol • Logische Uhrensynchronisation: Lamport-Timestamps und Vector-Timestamps <p>Koordination in verteilten Systemen</p> <ul style="list-style-type: none"> • Algorithmen zu Mutual Exclusion • Algorithmen zu Voting und Election • Verteilte Transaktionen <p>Replikation in verteilten Systemen</p> <ul style="list-style-type: none"> • Grundlegende Begriff zu Daten- und Objektreplikation • Replikationsalgorithmen zur Leistungssteigerung • Replikationsalgorithmen zur Fehlertoleranz • Replikation bei Transaktionen <p>Middlware</p> <ul style="list-style-type: none"> • Die Common Objekt Request Broker Architecture (CORBA) • CORBA Components • DCOM und GLOBE als Alternativen zu CORBA • Web Services <p>Weitere ausgewählte, aktuelle Themen</p>
Nachfolgemodule	Seminar Kommunikationssysteme Bachelor Thesis

Titel	Einführung in Eingebettete Software
Cluster	Bachelor: Wahlpflicht Software und Kommunikation
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	Deutsch/Englisch
Turnus	WS
Veranstalter	Kowalewski
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Grundlagen Technische Informatik
Literatur	<p>Folien zur Vorlesung, Skript sowie als Ergänzung folgende Bücher:</p> <ul style="list-style-type: none"> • Marwedel: Eingebettete Systeme. 2003 • Bass, Clements: Software Architecture in Practice.

	<ul style="list-style-type: none"> • Douglass: Real-time UML
Lernziele	<p>Erwerb der folgenden Kenntnisse und Fähigkeiten:</p> <ul style="list-style-type: none"> • Kenntnis und Beherrschung moderner Softwaretechnik für eingebettete Systeme • Erwerb der Sensibilität für die besonderen qualitativen Anforderungen beim Entwurf eingebetteter Software
Lerninhalte	<ul style="list-style-type: none"> • Technologische Grundlagen eingebetteter Systeme (Grundstruktur, Mikrocontroller, Speicherprogrammierbare Steuerungen) • Besondere Anforderungen beim Entwurf eingebetteter Software • Lebenszyklusmodelle • Analyse von funktionalen und nichtfunktionalen Anforderungen • Architektorentwurf- und analyse • Architekturelemente (Betriebssysteme, Busse, Middleware) • Modellierungs- und Analysetechniken für Verhalten und Struktur • Validierung (Simulation, Testen)
Nachfolgemodule	<ul style="list-style-type: none"> • Bachelor-Projekt

3.3 Daten- und Informationsmanagement

Modul	Implementation of Databases	
Semester	Bachelor-Wahlpflichtfach	
Dozent(in)	Prof. Dr. rer. pol. Matthias Jarke	
Sprache	English	
Lehrformen	V3 / Ü1	
Credit(s)	6 (180h, davon 60h Präsenz und 120h Eigenstudium)	
Voraussetzungen	<ul style="list-style-type: none"> • Introduction to Databases • Working knowledge in data structures 	
Lernziele/Kompetenzen	<p><i>General / Related to the modul:</i> The course offers an introduction to database architectures, query processing and optimization, transaction management, recovery, and administration of databases</p> <p><i>Subject-/Methodical-/Learning Competence/Soft Skills:</i> Students learn to analyse and optimize database structures and functionalities. In the exercises the students have to present their handed-in</p>	

	<p>solution in front of the class. Exercises can be done in small groups.</p> <p><i>Benefits for future professional life:</i> Professional knowledge about evaluating, administrating and tuning existing databases as well as a solid understanding of information system architectures in modern businesses is provided</p>
Inhalt	The module discusses the key aspects of the implementation of database systems. This includes the introduction of basic architectures (e.g. layered architecture) as well the procedures necessary for solving individual tasks (especially query analysis and transaction management). The concepts of implementation will be applied to classical (relational model, network model) as well as to more recent data models (distributed, object-oriented, deductive). In addition to the necessary theoretical background practical concepts will be introduced that allow database administrators the efficient tuning of databases.
Studienleistungen	<ul style="list-style-type: none"> • Active participation in lecture and exercises, submitted homeworks • Written exam at the end
Medienformen	<ul style="list-style-type: none"> • Lecture script in form of printed slides • Web page with exercises: http://www-i5.informatik.rwth-aachen.de/lehrstuhl/lehre/IDB03/index.html,
Literatur	<ul style="list-style-type: none"> • D.E. Shasha: Database Tuning - A Principled Approach. Prentice Hall, 1992. • R. Elmasri, S. Navathe: Fundamentals of Database Systems, Addison-Wesley, 4. Aufl. 2003 • T. Härder, E. Rahm: Datenbanksysteme – Konzepte und Techniken der Implementierung. Springer 1999. • G. Vossen: Datenmodelle, Datenbanksprachen und Datenbank-Management-Systeme. Addison-Wesley, 4. Aufl. 2004.

Titel	Introduction to Artificial Intelligence
Cluster	Bachelor Informatik, 5.-6. Semester
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
SWS	V3+Ü2
Sprache	English
Turnus	Every Fall Semester, make-up exam before the beginning of the next term
Veranstalter	Lakemeyer
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	
Literatur	<ul style="list-style-type: none"> • Lecture Notes (Transparencies) • Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach (2nd Edition), Addison Wesley, 2002.
Lernziele	<ul style="list-style-type: none"> • Students will gain knowledge in some of the major techniques

	developed in Artificial Intelligence. At the end of the course they will understand many of the basic ingredients that make up an intelligent agent enabling them to build such agents themselves.
Lerninhalte	<ul style="list-style-type: none"> • Agent Architecture • Heuristic Search • Games • Knowledge Representation • Machine Learning
Nachfolgemodule	Introduction to Knowledge Representation

Titel	Introduction to Knowledge Representation
Cluster	Bachelor Informatik, 5.-6. Semester
Credits	6
SWS	V3+Ü2
Sprache	English
Turnus	Every other Summer Semester, make-up exam before the beginning of the next term
Veranstalter	Lakemeyer
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Mathematical Logic
Literatur	<ul style="list-style-type: none"> • Lecture Notes (Transparencies) • Ron Brachman and Hector J. Levesque, Knowledge Representation and Reasoning, Morgan Kaufmann, 2004.
Lernziele	<ul style="list-style-type: none"> • Students will gain knowledge in some of the major techniques developed in Knowledge Representation. At the end of the course they will understand what distinguishes a knowledge-based system from others and they will be familiar with some of the main representation formalisms and reasoning techniques.
Lerninhalte	<ul style="list-style-type: none"> • First-Order Logic • Resolution • Horn Logic • Procedural Representations • Description Logics • Inheritance Networks

Modul	Data Mining Algorithms
Cluster	Daten- und Informationsmanagement
Studiengang, Semester	Bachelor Informatik, 4.-6. Semester
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3 Ü2
Sprache	Englisch
Turnus	Jährlich im Sommersemester
Veranstalter	Lehrstuhl für Informatik 9, Prof. Seidl
Leistungen	Klausurarbeit oder mündliche Prüfung
Voraussetzungen	Kenntnisse aus dem Modul <i>Datenstrukturen und Algorithmen</i> ; empfohlen sind Kenntnisse aus dem Modul <i>Datenbanken und Informationssysteme</i>
Literatur	Folienskript zur Vorlesung mit Verweisen auf Originalartikel sowie <ul style="list-style-type: none"> - Jiawei Han and Micheline Kamber: <i>Data Mining – Concepts and Techniques</i>. Morgan Kaufmann Publishers, 2000. - Martin Ester and Jörg Sander: <i>Knowledge Discovery in Databases – Techniken und Anwendungen</i>. Springer Verlag, 2000 (in German).
Lernziele	Erwerb der folgenden Kenntnisse und Fähigkeiten <ul style="list-style-type: none"> - Kenntnis grundlegender Konzepte und Methoden des Data Mining für große Datenbanken. - Kenntnis der Funktionalität und Leistungsfähigkeit von Algorithmen zum Data Mining. - Fähigkeit, Data Mining-Lösungen für konkrete Anwendungen zu bewerten
Lerninhalte	Concepts and Techniques for Data Mining <ol style="list-style-type: none"> 1. Introduction: KDD process, data mining tasks 2. Data warehousing and data preprocessing 3. Clustering: partitioning methods, density-based clustering, hierarchical clustering, subspace clustering, etc. 4. Classification: decision trees, nearest neighbor classifier, Bayes classifier, etc. 5. Mining association rules: Apriori-algorithm etc. 6. Generalization and concept description 7. Mining complex types of data
Nachfolgemodule	

Modul	Web Engineering
Cluster	B.Sc.: Praktische Informatik (Wahlpflicht)
Studiengang und Semester	B. Sc. Informatik, 5.-6. Semester
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V2+Ü3
Sprache	Deutsch/Englisch
Turnus	Wintersemester, ca. alle 2 Jahre
Veranstalter und Modulverantwortlicher	Schroeder (Lehr- und Forschungsgebiet Informatik 9 – Computerunterstütztes Lernen)
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Beherrschung der wesentlichen Konzepte imperativer und objektorientierter Programmiersprachen und Programmieretechniken • Kenntnisse im Bereich Software Engineering • Fähigkeit zur selbständigen Entwicklung kleinerer und mittlerer Programme
Literatur	<p>Folien und Skripte zur Vorlesung sowie z.B. folgende Bücher:</p> <ul style="list-style-type: none"> • Kappel, Reich, Retschitzegger: Web Engineering - Systematische Entwicklung von Web-Anwendungen, dpunkt.verlag, 2004 • Wöhr: Web Technologien, dpunkt, 2004 • Meinel/Sack: WWW, Springer 2004 <p>Weitere ausgewählte Kapitel aus diversen Büchern zu den jeweils vorgestellten Technologien.</p>
Lernziele	<p>Erwerb der folgenden Kenntnisse und Fähigkeiten:</p> <ul style="list-style-type: none"> • Kenntnis der relevanten Web-Technologien • Kenntnis des Prozess der Web-Anwendungsentwicklung • Fähigkeiten zur Beurteilung und Auswahl problem-adäquater Technologien und Werkzeuge für Web-Anwendungen
Lerninhalte	<p>Grundlagen der Web Technologien</p> <ul style="list-style-type: none"> • Client/Server Modelle, Protokolle, HTTP • Markup, HTML, CSS • XML, XSLT, XML Programmierung <p>Clientseitige Webprogrammierung</p> <ul style="list-style-type: none"> • Scripting, JavaScript, Flash • CGI, PHP, Datenbankanbindung <p>Web Engineering</p> <ul style="list-style-type: none"> • Vorgehensmodelle, Testen <p>Weitere aktuelle, ausgewählte Themen</p>
Nachfolgemodule	eLearning Seminar

	Bachelor Thesis im Bereich eLearning
--	--------------------------------------

3.4 Angewandte Informatik

Titel	Einführung in die Computergraphik
Cluster	Wahlpflicht/Computergraphik
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V2+Ü3
Sprache	Deutsch/Englisch
Turnus	Wintersemester (Wiederholungsprüfung im SS)
Veranstalter	Prof. Dr. L. Kobbelt
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Algorithmen und Datenstrukturen • Lineare Algebra
Literatur	<ul style="list-style-type: none"> • Foley, van Dam, Feiner, Hughes Computer Graphics: Principles and Practice • Watt 3D Computer Graphics
Lernziele	<ul style="list-style-type: none"> • Kenntnis der wichtigsten Datenstrukturen zur Darstellung von dreidimensionalen Objekten und Szenenbeschreibungen • Erlernen der elementaren Operationen und Methoden zur Transformation eines 3D Modells in ein realistisches zweidimensionales Bild (Rendering-Pipeline) • Verständnis der Graphik-API „OpenGL“ und die Fähigkeit, einfache Rendering-Techniken zu implementieren.
Lerninhalte	<ul style="list-style-type: none"> • Grundlagen der Geometriedarstellung (Polygonnetze, Volumendarstellungen, Freiform Kurven und Flächen) • Lokale Beleuchtung (3D Transformationen, Clipping, Rasterisierung, Lighting, Shading) • Globale Beleuchtung (Sichtbarkeitsproblem, Schattenberechnung, Ray Tracing) • Aufbau und Verwendung von „OpenGL“ • Performance-Optimierung von Graphik-Programmen
Ggf. Nachfolgemodule	<ul style="list-style-type: none"> • Basic Techniques in Computer Graphics • Geometry Processing • Globale Beleuchtung und Image-based Rendering • Computer Vision • Polynomial Curves and Surfaces • Subdivision Curves and Surfaces • Bachelor-Arbeit

Titel	Designing Interactive Systems I
Cluster	Bachelor-Wahlpflicht „Praktische Informatik“ (5.-6. Sem.)
Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V3+Ü2
Sprache	English
Turnus	Each WS
Veranstalter	Borchers
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	Required courses from the first four semesters
Literatur	D. Norman: <i>The Design Of Everyday Things</i> , Basic Books 2002 (required textbook for first few weeks), plus excerpts from <ul style="list-style-type: none"> • A. Dix et al.: <i>Human-Computer Interaction</i>, Prentice-Hall 2004 • B. Shneiderman et al.: <i>Designing The User Interf.</i>, Add.-W. 2004 • J. Raskin: <i>The Humane Interface</i>, Addison-Wesley 2000
Lernziele	After this class, students will know how user interfaces have developed over the past decades, and what constants of human performance need to be considered when designing them. They will be able to apply iterative design, prototyping, and evaluation methods to design usable, appropriate user interfaces in a user-centered fashion. All assignments are group assignments to foster collaboration skills, and project-based to strengthen project planning, conflict management and presentation skills. Students learn to think in designers' terms. This is a crucial competence for computer scientists working on user interfaces, a job that requires collaboration in interdisciplinary teams.
Lerninhalte	This class introduces students to human-computer interaction (HCI) and user interface design. It covers the following topics: <ul style="list-style-type: none"> • Fundamental characteristics of human cognition, such as reaction time, rules of perception, and memory performance • Models of interaction between people and their environment, such as affordances, mappings, constraints, slips and mistakes • Milestones in the history of human-computer interaction • Principles of iterative design • User interface prototyping techniques • Golden rules of user interface design • User interface design notations • User studies and evaluation methods
Nachfolgemodule	All follow-up modules are intended for the Master's level: <ul style="list-style-type: none"> • Designing Interactive Systems II • HCI Design Patterns • Current Topics in HCI

Titel	Introduction to High-Performance Computing
-------	---

Cluster	Angewandte Informatik
Credits	6 (180h, davon 60 h Präsenz und 120 h Eigenstudium)
Lehrformen/SWS	V3+Ü1
Sprache	Deutsch oder Englisch
Turnus	Jährlich
Veranstalter	Bischof , Bücken
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> Beherrschung der wesentlichen Konzepte imperativer und objektorientierter Programmiersprachen sowie elementarer Programmieretechniken in diesen Sprachen (Vorlesung Programmierung)
Literatur	<p>Handouts zur Vorlesung sowie z.B.:</p> <ul style="list-style-type: none"> V. Kumar, A. Grama, A. Gupta, G. Karypis: <i>Introduction to Parallel Computing: Design and Analysis of Algorithms</i>, 2nd Edition, Addison Wesley, 2003.
Lernziele	<p>Erwerb der folgenden Kenntnisse und Fähigkeiten:</p> <ul style="list-style-type: none"> Verständnis der wesentlichen Parallel-Rechnerarchitekturen Kenntnis grundlegender Entwurfsmethoden für datenlokale serielle und parallele Algorithmen Beherrschung einfacher Methoden zur Laufzeitanalyse von parallelen Algorithmen Grundlegendes Verständnis für elementare Operationen der parallelen Programmierung.
Lerninhalte	<ul style="list-style-type: none"> Parallele Rechnerarchitekturen, Netzwerk-Topologien Blockalgorithmen zur Ausnutzung von Datenlokalität in tiefen Speicherhierarchien Prinzipien des parallelen Algorithmenentwurfs Modellierung von Parallelität (Speedup, Effizienz, Amdahl) Einführung in parallele Programmierung Weitere ausgewählte Themen
Nachfolgemodule	

Titel	Computational Differentiation
Cluster	Angewandte Informatik
Credits	6 (180h, davon 60 h Präsenz und 120 h Eigenstudium)
Lehrformen/SWS	V3+Ü1
Sprache	Deutsch oder Englisch

Turnus	Jährlich
Veranstalter	Bischof, Bückler, Naumann
Leistungen	Klausurarbeit oder mündliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Beherrschung der wesentlichen Konzepte imperativer und objektorientierter Programmiersprachen sowie elementarer Programmieretechniken in diesen Sprachen (Vorlesung Programmierung) • Kenntnis elementarer diskreter Strukturen, insbesondere Graphen (Vorlesung Diskrete Strukturen)
Literatur	Handouts zur Vorlesung sowie z.B.: <ul style="list-style-type: none"> • A. Griewank: <i>Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation</i>, SIAM, 2000.
Lernziele	Erwerb der folgenden Kenntnisse und Fähigkeiten: <ul style="list-style-type: none"> • Beherrschung einfacher und fortgeschrittener Methoden zum automatischen Differenzieren • Verständnis für Laufzeit und Speicherbedarf von Algorithmen zum automatischen Differenzieren • Fähigkeit der Auswahl geeigneter Methoden des automatischen Differenzierens bei einer gegebenen Problemstellung • Grundlegendes Verständnis für die Umkehrung von Programmen.
Lerninhalte	<ul style="list-style-type: none"> • Vorwärts- und Rückwärtsmodus • Ausnutzung von Struktur (Dünnbesetztheit, Schnittstellenkontraktion) • Checkpointing • Parallelität in Ableitungsberechnungen • Modellierung durch Graphen • Weitere ausgewählte Themen
Nachfolgemodule	

4. Modulkatalog Anwendungsfächer (Anlage 4 zur BPO)

Im Folgenden sind die drei Fächer Betriebswirtschaftslehre, Elektrotechnik und Mathematik als „Pilot“-Anwendungsfächer beschrieben. Es ist vorgesehen, möglichst zügig weitere Anwendungsfächer per Prüfungsausschussbeschluss in das Modulhandbuch aufzunehmen.

4.1 Betriebswirtschaftslehre

Titel	Grundzüge der Wirtschaftswissenschaften
Cluster	Anwendungsfach Betriebswirtschaftslehre

Credits	6 (180h, davon 75h Präsenz und 105h Eigenstudium)
Lehrformen/SWS	V2+Ü2
Sprache	Deutsch
Turnus	Jedes WS, Wiederholungsprüfung im WS
Veranstalter	Dozenten des FB 8 (Fakultät für Wirtschaftswissenschaften)
Leistungen	Schriftliche Prüfung
Vorwissen	
Literatur	wird in der jeweiligen Veranstaltung bekanntgegeben
Lernziele	<ul style="list-style-type: none"> • Kenntnis der Prinzipien ökonomischen Denkens und Arbeitens • Kenntnis wichtiger wirtschaftswissenschaftlicher Denkschulen und Vorgehensweisen • Beherrschung der wirtschaftswissenschaftlichen Terminologie • Kenntnis zentraler mikro- und makroökonomischer Probleme und grundlegender Konzepte zu ihrer Lösung • Grundlagenkenntnisse wichtiger wirtschaftswissenschaftlicher Methoden und Konzepte • Fähigkeit zum kritisch-reflektierenden Herangehen an wirtschaftliche Fragestellungen
Lerninhalte	<ul style="list-style-type: none"> • Merkmale ökonomischen Denkens • Kennzeichnung, Abgrenzung und Zusammenhänge zwischen Betriebs- und Volkswirtschaftslehre, Mikro- und Makroökonomie sowie Betriebs- und Volkswirtschaftstheorie und -politik • Wichtige wirtschaftswissenschaftliche Methoden und Instrumente • Kennzeichnung, Analyse und Lösungsansätze zentraler wirtschaftswissenschaftlicher Fragestellungen
Nachfolgemodule	Quantitative Methoden (Operations Research), Entscheidungslehre, Grundzüge des betrieblichen Rechnungswesens

Titel	Grundzüge des betrieblichen Rechnungswesens
Cluster	Anwendungsfach Betriebswirtschaftslehre
Credits	4 (120h, davon 45h Präsenz und 75h Eigenstudium)
Lehrformen/SWS	V2+Ü1
Sprache	Deutsch
Turnus	Jedes SS (erstmalig im SS 2009), Erst- und Wiederholungsprüfung in der vorlesungsfreien Zeit des SS
Veranstalter	Prof. Hömberg und Mitarbeiter
Leistungen	schriftliche Prüfung
Vorwissen	Kein Vorwissen erforderlich

Literatur	<ul style="list-style-type: none"> • Baetge/Kirsch/Thiele: Bilanzen, 8. Aufl., Düsseldorf: IDW-Verlag, 2005 • Coenenberg, Adolf G./Mattner, Gerhard/Schultze, Wolfgang: Einführung in das Rechnungswesen, Grundzüge der Buchführung und Bilanzierung, Stuttgart: Schäffer-Poeschel, 2004 • Möller, Hans Peter/Zimmermann, Jochen/Hüfner, Bernd: Erlös- und Kostenrechnung, München u.a.: Pearson, 2005 • Schmidt, Andreas: Kostenrechnung, Grundlagen der Vollkosten-, Deckungsbeitrags- und Plankostenrechnung sowie des Kostenmanagements, 4. Aufl., Stuttgart u.a.: Kohlhammer, 2005
Lernziele	<p>Nach erfolgreichem Absolvieren sollen die Studierenden kennen und beurteilen können:</p> <ul style="list-style-type: none"> • Grundbegriffe des Rechnungswesens, • Aufgaben und zentrale rechtliche Vorschriften des Rechnungswesens, • Methoden des Rechnungswesens, • Anwendungsprobleme
Lerninhalte	<ul style="list-style-type: none"> ○ Abbildung vom Geschäftsvorfällen in Buchhaltung und Jahresabschluß ○ Grundzüge des Externen Rechnungswesens ○ Grundzüge des Internen Rechnungswesens

Titel	Entscheidungslehre (Wirtschaftswissenschaften C)
Cluster	Anwendungsfach Betriebswirtschaftslehre
Credits	6 (180h, davon 60h Präsenz und 120h Eigenstudium)
Lehrformen/SWS	V2+Ü2
Sprache	Deutsch
Turnus	Jedes WS zwei Termine
Veranstalter	Prof. Dr. Rüdiger von Nitzsch
Leistungen	schriftliche Prüfung
Vorwissen	<ul style="list-style-type: none"> • Grundkenntnisse BWL • Gesunder Menschenverstand
Literatur	<p>Grundlegende Literatur</p> <ul style="list-style-type: none"> • Von Nitzsch, R. (2002), Entscheidungslehre, Schäffer-Poeschel, Stuttgart • Von Nitzsch, R. (2005), Anwendungen der Entscheidungslehre: Bewertung betrieblicher Investitionen, Skript (erhältlich am Lehrstuhl ABWL während der Öffnungszeiten) <p>Weiterführende Literatur</p> <ul style="list-style-type: none"> • Bamberg, G. & Coenenberg, A.G. (2004), Betriebswirtschaftliche Entscheidungslehre, Vahlen • Eisenführ, F. & Weber, M. (2002), Rationales Entscheiden, 4.

	<p>Auflage, Springer</p> <ul style="list-style-type: none"> • Jungermann, H. Pfister, H.R. & Fischer, K. (2005), Die Psychologie der Entscheidung, 2. Auflage, Spektrum
Lernziele	<ul style="list-style-type: none"> • Strukturieren von komplexen Entscheidungsproblemen • Lösungsansätze für Entscheidungsprobleme • Vermeidung psychologischer Fallen bei Entscheidungen
Lerninhalte	<ul style="list-style-type: none"> • Instrumente der präskriptiven Entscheidungslehre • Anwendung: Bewertung betrieblicher Investitionen • Die Psychologie der Entscheidung

Titel	Quantitative Methoden (Operations Research)
Cluster	Anwendungsfach Betriebswirtschaftslehre
Credits	6 (180h, davon 60h Präsenz und 120h Eigenstudium)
Lehrformen/SWS	V2+Ü2
Sprache	Deutsch
Turnus	Jährlich (jeweils im Sommersemester)
Veranstalter	Dozenten der Wirtschaftswissenschaften
Leistungen	Lösung von Übungsaufgaben, schriftliche Prüfung
Vorwissen	Keine über die generellen Anforderungen des Bachelor-Studienganges hinausgehenden Voraussetzungen
Literatur	<ul style="list-style-type: none"> • H.-J. Zimmermann, Operations Research Methoden und Modelle, Vieweg, 2005 • K. Neumann, M. Morlock, Operations Research, 2. Auflage, Hanser, 2002
Lernziele	<p>Nach erfolgreichem Absolvieren der Lehrveranstaltung werden die Studierenden</p> <ul style="list-style-type: none"> • die wichtigsten Grundlagen, Methoden und Algorithmen der Linearen Optimierung kennen, • in der Lage sein, Probleme aus der Produktionsplanung und Logistik (insbesondere Transport) als Lineare Optimierungsprobleme zu modellieren, • Probleme und Methoden zur Behandlung gemischt-ganzzahliger Optimierungsprobleme kennen und • in der Lage sein, spezielle lineare bzw. gemischt-ganzzahlige Optimierungsprobleme mit einer Modellierungssprache abzubilden und zu lösen

Lerninhalte	In der Lehrveranstaltung werden quantitative Methoden der Wirtschaftswissenschaften vorgestellt. Insbesondere werden Modelle, Methoden und Algorithmen behandelt, die eine besonders hohe Bedeutung für die Wirtschaftswissenschaften und für Anwendungen in der Praxis besitzen. Im Einzelnen werden Lineare Optimierung und eine Einführung in die Diskrete und Kombinatorische Optimierung behandelt.
-------------	--

4.2 Elektrotechnik

Die im Folgenden beschriebenen beiden Module Grundgebiete der Elektrotechnik A und B sind Pflichtbestandteil des Anwendungsfachs Elektrotechnik. Die noch zusätzlich zu erbringenden 4 Kreditpunkte können durch Wahl eines der folgenden Module aus dem Bachelor Elektrotechnik erworben werden. Die jeweiligen Detailangaben sind dem zurzeit entstehenden Modulhandbuch zu entnehmen:

- Kommunikationstechnik (Modulverantwortl.: Ohm oder Vary)
- Energieversorgung (Haubrich)
- Kommunikationsnetze (Walke)
- Elektronische Bauelemente (Heinen)
- Bildverarbeitung (Aach)
- Nachrichtensysteme (Vary)

Modul	Grundgebiete der Elektrotechnik A
Cluster	Nebenfach Elektrotechnik
Studiengang- und Semester	Bachelor Informatik 4. Semester
Credits	8
Lehrformen/SWS	V4+Ü2
Sprache	Deutsch
Turnus	Jährlich
Veranstalter	De Doncker
Leistungen	Klausurarbeit
Voraussetzungen	keine
Literatur	wird in der Vorlesung bekannt gegeben
Lernziele	Beherrschung der Grundlagen der Elektrotechnik in den Bereichen Wechselstromrechnung, Behandlung konzentrierter Bauelemente und Mehrphasensysteme mit dem Ziel, Zusammenhänge mit Anwendungen der Informatik zu erkennen und mit Elektrotechnikern als

	Informatikanwendern kompetent kommunizieren zu können.
Lerninhalte	<ul style="list-style-type: none"> • Darstellung von Wechselgrößen: Wechselstromkenngrößen, reelle Wechselstromrechnung, Zeigerdarstellung, komplexe Wechselstromrechnung, Leistungsbegriffe bei Wechselgrößen; • Konzentrierte Elemente: Grundlagen und Bauformen der konzentrierten Elemente R, C, L, allgemeine Systemgleichungen, Schaltvorgänge an den konzentrierten Elementen, stationäre harmonische Betrachtung, stationäre und transiente Vorgänge an RC- und RL- Gliedern, Schwingkreise, Bodediagramm, Transformator; • Mehrphasensysteme: elektromechanische und leistungselektronische Erzeugung von Mehrphasensystemen, Analyse symmetrischer Drehstromnetzwerke, Nichtlineare Bauteile und Schaltungen: der reale Transformator, Hysterese- und Wirbelstromverluste, nichtlineare Eigenschaften magnetischen Materials, Gleichrichterschaltungen, Linearregler, Schaltnetzteile, Batterien. • Leitungsgleichungen für stationären Betrieb, Grundlagen Gleichstrommotor
Nachfolgemodule	Grundgebiete der Elektrotechnik B

Modul	Grundgebiete der Elektrotechnik B
Cluster	Nebenfach Elektrotechnik
Studiengang- und Semester	Bachelor Informatik 5. Semester
Credits	8
Lehrformen/SWS	V4+Ü2
Sprache	Deutsch
Turnus	Jährlich
Veranstalter	Ohm
Leistungen	Klausurarbeit
Voraussetzungen	keine
Literatur	wird in der Vorlesung bekannt gegeben
Lernziele	Beherrschung der Grundlagen der Analyse von Signalen und Systemen mit dem Ziel, Zusammenhänge mit Anwendungen der Informatik zu erkennen und mit Elektrotechnikern als Informatikanwendern kompetent kommunizieren zu können.
Lerninhalte	<ul style="list-style-type: none"> • Analyse instationärer Vorgänge, Stationäre Anregung mit Wechselspannungsquellen, Geschaltete Gleichspannungsquellen, Anregung mit geschalteten Wechselspannungsquellen; • Signale und Systeme: Elementarsignale, Begriff des Systems,

	<p>Lineare zeitinvariante Systeme, Das Faltungsintegral, Beispiel zur Berechnung des Faltungsintegrals, Faltungsalgebra, Dirac-Impuls, Integration und Differentiation von Signalen, Kausale und stabile Systeme, Energie und Leistung von Signalen;</p> <ul style="list-style-type: none"> • Fourieranalyse: Eigenfunktionen von LTI-Systemen, Fourierreihen, Das Fourier-Integral, Theoreme zur Fourier-Transformation, Beispiele zur Anwendung der Theoreme, Tabellen zur Fourier-Transformation; • Zeit- und Frequenzverhalten von Signalen und Systemen: Das verzerrungsfreie System, Parameter zur Charakterisierung von Übertragungseigenschaften, Tiefpasssysteme, Hochpass- und Bandpasssysteme; • Laplace-Transformation: Konvergenzbetrachtungen zur Fourier- und Laplace-Transformation, Beispiele zur Laplace-Transformation, Pole und Nullstellen in der komplexen Laplace-Ebene, Inverse Laplace-Transformation, Lösung von Differentialgleichungen mittels der Laplace-Transformation, Stabilitätsanalyse von Systemen, Systemanalyse und -synthese mittels der Laplace-Transformation, Tabellen zur Laplace-Transformation; • Zeitdiskrete Signale und Systeme: Abtastung im Zeitbereich, Zeitdiskrete Signale und Systeme, Diskrete Faltung, Zeitdiskrete Elementarsignale, Lineare verschiebungsinvariante Systeme, Beispiel zur diskreten Faltung, Fourier-Transformation zeitdiskreter Signale, Die diskrete Fourier-Transformation, z-Transformation, Zeitdiskrete Tief-, Band- und Hochpasssysteme, Tabellen zur Fourier- und z-Transformation diskreter Signale; • Leitungstheorie: Stationäre Lösung der Wellengleichung, Wellenausbreitung auf Leitungen; • Korrelationsanalyse : Energie- und Leistungssignale – Orthogonalität, Kreuzkorrelation, Autokorrelation, Faltung und Energiedichtespektrum – Korrelationsanalyse zeitdiskreter Signale; <p>Statistische Signalbeschreibung: Zufallssignale – Stationarität und Ergodizität – Mittelwerte, Korrelationsfunktionen, Momente und Leistungsdichtespektren stationärer Prozesse – Zufallssignale in LTI-Systemen, Weißes Rauschen – Verteilungs- und Verteilungsdichtefunktionen – Gauß-Verteilungen –zeitdiskrete Zufallssignale</p>
--	--

4.3 Mathematik

Um die für das Anwendungsfach Mathematik vorgesehenen 22 Kreditpunkte zu erzielen, ist aus den Modulen *Algebra* und *Funktionentheorie I* (jeweils 10 Kreditpunkte) eins auszuwählen. Dazu können entweder das *Mathematische Praktikum* (6 Kreditpunkte) und *Numerische Analysis I* (6 Kreditpunkte) oder *Numerische Analyse I und II* (jeweils 6 Kreditpunkte genommen werden).

Modul:	Algebra
Verantwortlich:	Prof. Dr. G. Hiß, Prof. Dr. W. Plesken
Weitere Dozenten:	Prof. Dr. G. Nebe, Prof. Dr. E. Zerz
Lernziele:	Die Studierenden sollen vertieftes Verständnis für algebraische Strukturen wie Gruppen, Ringe, Moduln, Körper erwerben, das Zusammenspiel algebraischer Begriffsbildungen kennen lernen, an mindestens einem Beispiel eine Strukturtheorie vertiefen, den Bezug der Algebra zu anderen Disziplinen entdecken und Grundwissen für weitere algebraische Studien erwerben.
Modulinhalte:	Strukturtheorie endlicher Gruppen, halbeinfache Algebren und ihre Darstellungen, Galoistheorie
Einordnung:	Vertiefungsmodul im Bachelor-Studiengang im 5. oder 6. Semester
Lehrform/SWS:	Vorlesung (4 SWS), Übung (2 SWS)
Sprache:	Deutsch
Literatur:	M. Artin: Algebra, Birkhäuser 1993 S. Lang: Algebra (third edition), Addison Wesley 1995
Prüfungsleistungen:	Klausurarbeit
Arbeitsaufwand:	300 Stunden, davon 84 Stunden Präsenz
Kreditpunkte:	10
Dauer des Moduls:	Ein Semester
Häufigkeit des Angebots:	Jedes Wintersemester

Modul:	Funktionentheorie I
Verantwortlich:	Prof. Dr. A. Krieg
Weitere Dozenten:	Prof. Dr. G. Jank, Prof. Dr. S. Walcher
Lernziele:	Die Studierenden sollen die Grundzüge der komplexen Analysis beherrschen und ihre Bedeutung für die reelle Analysis kennen lernen.
Modulinhalte:	Komplexe Differenzierbarkeit und Cauchy-Riemannsche Differentialgleichungen, Kurvenintegrale, Cauchysche Theorie, Abbildungsverhalten holomorpher Funktionen, einfach zusammenhängende Gebiete, isolierte Singularitäten, Residuensatz mit Anwendungen auf reelle Integrale, Produktdarstellungen, Riemannscher Abbildungssatz.
Einordnung:	Aufbaumodul 5. Semester (W-Variante) bzw. 6. Semester (S-Variante)
Lehrform/SWS:	Vorlesung (4 SWS), Übung (2 SWS)
Sprache:	Deutsch
Literatur:	W. Fischer, I. Lieb: Funktionentheorie, Vieweg 2005

	E. Freitag, W. Busam: Funktionentheorie, Springer-Verlag, Berlin 2000 A. Krieg: Analysis IV, Skript, RWTH Aachen 2005 R. Remmert, G. Schumacher: Funktionentheorie, Springer-Verlag, Berlin 2002
Prüfungsleistungen:	Klausurarbeit
Arbeitsaufwand:	300 Stunden, davon 84 Stunden Präsenz
Kreditpunkte:	10
Dauer des Moduls:	Ein Semester
Häufigkeit des Angebots:	Jedes Wintersemester

Modul:	Mathematisches Praktikum
Verantwortlich:	Prof. Dr. W. Dahmen
Weitere Dozenten:	Prof. Dr. H. Esser, Priv.-Doz. Dr. H. Jarausch, Priv.-Doz. Dr. S. Müller, Prof. Dr. S. Noelle, Prof. Dr. A. Reusken
Lernziele:	Die Studierenden sollen lernen, für Probleme aus verschiedenen Gebieten der Mathematik effiziente algorithmische Lösungen zu entwickeln. Sie sollen die Fähigkeit zur Umsetzung abstrakter Algorithmen in C++ Programme erwerben, Grundlagen erarbeiten, um Programmieraufgaben für andere mathematische Veranstaltungen des Bachelor-Studiums zu lösen, und Voraussetzungen schaffen, um später bei der mathematischen Simulation naturwissenschaftlicher und technischer Probleme mitzuwirken.
Modulinhalte:	Wechselnde Fragestellungen und Algorithmen aus der diskreten Optimierung, Gruppentheorie, Zahlentheorie, linearen Algebra, Bildverarbeitung, Datenkompression, Numerik etc.
Einordnung:	Fachübergreifendes Modul im 3. oder 4. Semester
Lehrform/SWS:	Beratung/Diskussion (2 SWS), Rechnerübung (2 SWS)
Sprache:	Deutsch
Literatur:	Wechselnd je nach behandelten Themen
Prüfungsleistungen:	Regelmäßige Teilnahme und Testate für Programmieraufgaben
Arbeitsaufwand:	180 Stunden, davon 56 Stunden Präsenz
Kreditpunkte:	6
Dauer des Moduls:	Ein Semester
Häufigkeit des Angebots:	Jedes Semester

Modul:	Numerische Analysis I
Verantwortlich:	Prof. Dr. W. Dahmen, Prof. Dr. A. Reusken
Weitere Dozenten:	Prof. Dr. H. Esser, Prof. Dr. S. Noelle
Lernziele:	Die Studierenden sollen Verständnis für grundlegende Begriffe der numerischen Analysis, insbesondere der Kondition eines Problems

	und Stabilität eines Algorithmus und der darauf basierenden Fehleranalyse entwickeln, die Fähigkeit erwerben, grundlegende numerische Methoden in ihrer Funktionsweise zu verstehen, die durch sie erreichbaren Ergebnisse einzuschätzen und darauf aufbauend in flexibler Anpassung an neue Aufgabenstellungen die Methode weiter zu entwickeln, die Grundbegriffe und Konzepte wie Matrixfaktorisierungen, Projektionen und iterative Lösungsansätze sicher beherrschen und die Fähigkeit zum aktiven Umgang mit den Gegenständen der Lehrveranstaltung erwerben und aufbauend auf diesen methodischen Werkzeugen erste grundlegende Konzepte für das approximative Lösen wissenschaftlicher und technischer Probleme aneignen.
Modulinhalte:	Fehleranalyse, Kondition, Rundungsfehler, Stabilität. Direkte Lösungsverfahren für lineare Gleichungssysteme. Lineare Ausgleichsrechnung. Iteratives Lösen nicht-linearer Gleichungssysteme. Nichtlineare Ausgleichsrechnung. Lösen von Eigenwertproblemen.
Einordnung:	Grundmodul 3. Semester (W-Variante) bzw. 4. Semester (S-Variante)
Lehrform/SWS:	Vorlesung (2 SWS), Übung (2 SWS), Kleingruppenübung
Sprache:	Deutsch
Literatur:	W. Dahmen, A. Reusken, Numerik für Ingenieure und Naturwissenschaftler, Springer 2006. P. Deuflhard, A. Hohmann, Numerische Mathematik I, de Gruyter 2002. A. Reusken, Numerische Analysis I (Skript).
Prüfungsleistungen:	Klausurarbeit oder mündliche Prüfung
Arbeitsaufwand:	180 Stunden, davon 56 Stunden Präsenz
Kreditpunkte:	6
Dauer des Moduls:	Ein Semester
Verwendbarkeit:	Voraussetzung für Numerische Analysis II
Häufigkeit des Angebots:	Jedes Wintersemester

Modul:	Numerische Analysis II
Verantwortlich:	Prof. Dr. W. Dahmen, Prof. Dr. A. Reusken
Weitere Dozenten:	Prof. Dr. H. Esser, Prof. Dr. S. Noelle
Lernziele:	Die Studierenden sollen das Verständnis für grundlegende Begriffe der numerischen Analysis, insbesondere Kondition eines Problems und Stabilität eines Algorithmus sowie der darauf basierenden Fehleranalyse, vertiefen, die Fähigkeit erwerben, grundlegende numerische Methoden in ihrer Funktionsweise zu verstehen, die durch sie erreichbaren Ergebnisse einzuschätzen und darauf aufbauend in flexibler Anpassung an neue Aufgabenstellungen die Methode weiter zu entwickeln, Grundbegriffe und -techniken wie Interpolation, Glattheits-Eigenschaften und Approximationsgüte sicher beherrschen und die Fähigkeit zum aktiven Umgang mit den Gegenständen der Lehrveranstaltung erwerben und aufbauend auf

	diesen methodischen Werkzeugen erste grundlegende Konzepte für das approximative Lösen wissenschaftlicher und technischer Probleme aneignen.
Modulinhalte:	Approximation und Interpolation mit Polynomen. Spline-Funktionen. Schnelle Fourier-Transformation. Numerische Integration.
Einordnung:	Grundmodul 4. Semester (W-Variante) bzw. 5. Semester (S-Variante)
Modulvoraussetzungen:	Kenntnisse der Module Analysis I, Lineare Algebra I, Numerische Analysis I.
Lehrform/SWS:	Vorlesung (2 SWS), Übung (2 SWS), Kleingruppenübung
Sprache:	Deutsch
Literatur:	W. Dahmen, A. Reusken, Numerik für Ingenieure und Naturwissenschaftler, Springer 2006. P. Deufhard, A. Hohmann, Numerische Mathematik I, de Gruyter 2002. A. Reusken, Numerische Analysis II (Skript).
Prüfungsleistungen:	Klausurarbeit oder mündliche Prüfung
Arbeitsaufwand:	180 Stunden, davon 56 Stunden Präsenz
Kreditpunkte:	6
Dauer des Moduls:	Ein Semester
Häufigkeit des Angebots:	Jedes Sommersemester