

Studienprogramm des Bachelor-Studiengangs Informatik

Stand August 2007



Inhaltsverzeichnis

Modulübersicht	5
Prüfungsordnung-Struktur	6
Konsequenzregelungen	8
Internationalisierung	9
Module des ersten Studienjahrs	10
1. Modul: Mathematische Grundlagen – Algebra	10
1.1 Unit: Vorlesung Mathematische Grundlagen – Algebra	11
1.2 Unit: Übung Mathematische Grundlagen – Algebra	12
2. Modul: Grundlagen der Informatik und maschinennahe Programmierung	12
2.1 Unit: Vorlesung Grundlagen der Informatik und maschinennahe Programmierung	13
2.2 Unit: Übung maschinennahe Programmierung	15
3. Modul: Einführung in die Programmierung mit C	16
3.1 Unit: Vorlesung Einführung in die Programmierung mit C	17
3.2 Unit: Übung Programmierung mit C	18
3.3 Unit: UNIX Praktikum	18
4. Modul: Betriebswirtschaftslehre	19
4.1 Unit: Vorlesung Betriebswirtschaftslehre	20
5. Modul: Englisch	21
5.1 Unit: Englisch 1	22
5.2 Unit: Englisch 2	22
6. Modul: Diskrete Mathematik	23
6.1 Unit: Vorlesung Diskrete Mathematik	24
6.2 Unit: Übung Diskrete Mathematik	25
7. Modul: Digitaltechnik	25
7.1 Unit: Vorlesung Digitaltechnik	26
7.2 Unit: Übung Digitaltechnik	27
8. Modul: Theoretische Grundlagen der Informatik	27
8.1 Unit: Vorlesung Rechnerarchitektur	28
8.2 Unit: Übung Rechnerarchitektur	29
8.3 Unit: Vorlesung Automatentheorie und formale Sprachen	29
8.4 Unit: Übung Automatentheorie und formale Sprachen	30
9. Modul: Algorithmen und Datenstrukturen	30
9.1 Unit: Vorlesung Algorithmen und Datenstrukturen	31
9.2 Unit: Übung Algorithmen und Datenstrukturen	33
10. Modul: Vorlesung Objektorientierte Programmierung Grundlagen und Konzepte – C++	33
10.1 Unit: Vorlesung Objektorientierte Programmierung Grundlagen und Konzepte –C++	34
10.2 Unit: Übung Objektorientierte Programmierung mit C++	35
Module des zweiten Studienjahrs	36
11. Modul: Software Engineering - Analyse	36
11.1 Unit: Vorlesung Software Engineering - Analyse	37
11.2 Unit: Übung Software Engineering - Analyse	38
12. Modul: Statistik	39
12.1 Unit: Vorlesung Statistik	40
12.2 Unit: Übung Statistik	41



13. Modul: Objektorientierte Programmierung Vertiefung - Java	41
13.1 Unit: Vorlesung Objektorientierte Programmierung Vertiefung -Java	42
13.2 Unit: Übung Objektorientierte Programmierung Java.....	43
14. Modul: Datenbanken	43
14.1 Unit: Vorlesung Datenbanken	44
14.2 Unit: Übung Datenbanken	45
15. Modul: Rechnernetze	46
15.1 Unit: Vorlesung Rechnernetze.....	47
15.2 Unit: Übung Rechnernetze	48
16. Modul: Betriebssysteme	48
16.1 Unit: Vorlesung Betriebssysteme	49
16.2 Unit: Übung Betriebssysteme	50
17. Modul: Numerische Mathematik	50
17.1 Unit: Vorlesung Numerische Mathematik.....	51
17.2 Unit: Übung Numerische Mathematik.....	52
18. Modul: Software Engineering – Design	52
18.1 Unit: Vorlesung Software Engineering – Design	54
18.2 Unit: Übung Software Engineering – Design.....	55
19. Modul: Echtzeitsysteme.....	56
19.1 Unit: Vorlesung Echtzeitsysteme	56
19.2 Unit: Übung Echtzeitlabor.....	57
20. Modul: Datenschutz / IT Security.....	58
20.1 Unit: Vorlesung Datenschutz.....	59
20.2 Unit: Vorlesung IT Security	60
21. Modul: IT Projekte	61
21.1 Unit: IT Projektmanagement	62
21.2 Unit: Moderation, Präsentation, Zeitmanagement.....	63
21.3 Unit: Fallstudie Prozessmodellierung.....	64
21.4 Unit: Programmierpraktikum mit Datenbanken	64
Module des dritten Studienjahrs	66
22. Modul: Verteilte Anwendungen	66
22.1 Unit: Vorlesung Verteilte Anwendungen	66
22.2 Unit: Übung Verteilte Anwendungen.....	67
23. Modul: Moderne Netzstrukturen	68
23.1 Unit: Vorlesung Moderne Netzstrukturen	69
23.2 Unit: Übung Moderne Netzstrukturen	69
24. Modul: Informatik Projekt	70
24.1 Unit: Projekt.....	71
25. Modul: Aktuelle Themen der Informatik	71
25.1 Unit: Vorlesung	72
25.2 Unit:Seminar.....	72
26. Modul: Studium Generale.....	73
27. Modul: Praxisphase	74
27.1 Unit: Seminar zur Praxisphase.....	76
27.2 Unit: Betreutes Praxisprojekt.....	76



28. Modul: Bachelorarbeit	77
Zusatzqualifikation „Betrieblicher Datenschutz“	79



Modulübersicht

Semester Module

1	Mathematische Grundlagen	Einführung in die Programmierung	Grundlagen der Informatik und maschinennahe Programmierung
	Betriebswirtschaftslehre		
2	Diskrete Mathematik	Theoretische Grundlagen der Informatik	Englisch (Englisch 1 1. Sem, Englisch 2 2. Sem.)
	Digitaltechnik	Objektorientierte Programmierung - Grundlagen, Konzepte	
3	Statistik	Objektorientierte Programmierung - Vertiefung	Betriebssysteme
	Software Engineering Analyse	Datenbanken	Rechnernetze
4	Numerische Mathematik	IT Projekte	Echtzeitsysteme
	Software Engineering Design		Datenschutz / IT Security
5	Verteilte Anwendungen	Informatik Projekt	Aktuelle Themen der Informatik
	Moderne Netzstrukturen		Studium Generale
6	Praxisphase		
	Bachelorarbeit mit Kolloquium		



Prüfungsordnung-Struktur

Sem.	Nr.	Modul	Unit	Lehrform	Art	Typ	SWS	CP	Workload
1	1	Mathematische Grundlagen - Algebra	Vorlesung Mathematische Grundlagen - Algebra	V	K	PL	4	5	150
			Übung Mathematische Grundlagen - Algebra	Ü			2		
	2	Grundlagen der Informatik und maschinennahe Programmierung	Vorlesung Grundlagen der Informatik und maschinennahe Programmierung	V			4	10	300
			Übung Maschinennahe Programmierung	L	A	VL	2		
	3	Einführung in die Programmierung mit C	Vorlesung Einführung in die Programmierung mit C	V	KR	PL	2	10	300
Übung Programmierung mit C			L	A	VL	2			
Unix Praktikum			L	A	VL	2			
4	Betriebswirtschaftslehre	Vorlesung Betriebswirtschaftslehre	V	K	PL	4	5	150	
5	Englisch	Englisch 1	Ü	A	VL	2		60	
							24	30	960
2	6	Diskrete Mathematik	Vorlesung Diskrete Mathematik	V	K	PL	4	5	150
			Übung Diskrete Mathematik	Ü			2		
	7	Digitaltechnik	Vorlesung Digitaltechnik	V	K	PL	2	5	150
			Übung Digitaltechnik	Ü	A	VL	2		
	8	Theoretische Grundlagen der Informatik	Vorlesung Automatentheorie und formale Sprachen	V			1	5	150
			Übung Automatentheorie und formale Sprachen	Ü			1		
			Vorlesung Rechnerarchitektur	V			1		
			Übung Rechnerarchitektur	Ü			1		
	9	Algorithmen und Datenstrukturen	Vorlesung Algorithmen und Datenstrukturen	V	K	PL	2	5	150
			Übung Algorithmen und Datenstrukturen	Ü			2		
10	Objektorientierte Programmierung Grundlagen und Konzepte -- C++	Vorlesung Objektorientierte Programmierung Grundlagen und Konzepte - C++	V			2	5	150	
		Übung Objektorientierte Programmierung mit C++	L			2			
		5	Englisch	Englisch 2	Ü				PL
							25	30	840
3	11	Software Engineering - Analyse	Vorlesung Software Engineering - Analyse	V	K	PL	2	5	150
			Übung Software Engineering - Analyse	L	B	VL	2		
	12	Statistik	Vorlesung Statistik	V	K	PL	2	5	150
			Übung Statistik	Ü			2		
	13	Objektorientierte Programmierung Vertiefung -- Java	Vorlesung Objektorientierte Programmierung Vertiefung -- Java	V			2	5	150
			Übung Objektorientierte Programmierung Java	L			2		
			14	Datenbanken	Vorlesung Datenbanken	V	K		
	Übung Datenbanken	L					2		
	15	Rechnernetze	Vorlesung Rechnernetze	V	K	PL	2	5	150
			Übung Rechnernetze	L	A	VL	2		
16	Betriebssysteme	Vorlesung Betriebssysteme	V	K	PL	2	5	150	
		Übung Betriebssysteme	L			2			
							26	30	900



4	17 Numerische Mathematik	Vorlesung Numerische Mathematik	V	K	PL	2	5	150	
		Übung Numerische Mathematik	L			2			
	18 Software Engineering - Design	Vorlesung Software Engineering - Design	V	K	PL	2	5	150	
		Übung Software Engineering - Design	L			2			
	19 Echtzeitsysteme	Vorlesung Echtzeitsysteme	V	K	PL	2	5	150	
		Übung Echtzeitsysteme	L	B	VL	2			
	20 Datenschutz / IT Security	Vorlesung Datenschutz	V	K	PL	2	5	150	
		Vorlesung IT Security	V			2			
	21 IT-Projekte	Vorlesung IT-Projektmanagement Moderation, Präsentation, Zeitmanagement	V	K	TPL	2	10	300	
			Ü	A	VL	2			
Fallstudie Prozessmodellierung Programmierpraktikum mit Datenbanken		P	P	TPL	2				
		P	P	TPL	2				
							24	30	900
5	22 Verteilte Anwendungen	Vorlesung Verteilte Anwendungen	V	K	PL	2	5	150	
		Übung Verteilte Anwendungen	L	B	VL	2			
	23 Moderne Netzstrukturen	Vorlesung Moderne Netzstrukturen	V	K	PL	2	5	150	
		Übung Moderne Netzstrukturen	Ü			2			
	24 Informatik-Projekt	Projekt	P	P	PL	4	10	300	
	25 Aktuelle Themen der Informatik	Seminar	S		TPL	2	5	150	
Vorlesung		V		TPL	2				
26 Studium Generale		V		PL	2	5	150		
							18	30	900
6	27 Praxisphase	Praxisphasenseminar	S	ABV	PL	2	18	540	
		betreutes Praxisprojekt (14 Wo a 4,5 Tage)	P	C	VL				
	28 Bachelorarbeit				PL		12	360	
							2	30	900

Legende:

Lehrformen: V=Vorlesung, Ü=Übung,
L=Labor, S=Seminar, P=Projekt

Leistungsnachweis: VL

Art: K = Klausur, T = Testate,

KR = Klausur am Rechner,
= Anwesenheit,

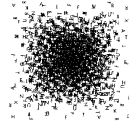
= (Programmier-)Projekt,
Anwesenheit, Bericht und Vortrag

Lehrformen	S				
	V	U	L	S	P
Summe	25	11	12	2	4


Module	PL	TPL	VL	SWS	CP	Work-load
28	26	5	11	119	180	5400

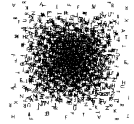
Modelle zur Erbringung der Vorleistungen #

A	Regelmäßige Teilnahme, das heißt Teilnahme an mindestens 80% aller	7
B	Bestehen aller angebotenen Testate	3
C	Bescheinigung der Praxisfirma und des betreuenden Professors über die	1



Konsequenzregelungen

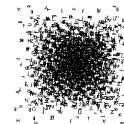
Sem. Modul	notwendige Voraussetzungen
1 Mathematische Grundlagen Einführung in die Programmierung Programmierung Betriebswirtschaftslehre Englisch	
2 Diskrete Mathematik Digitaltechnik Theoretische Grundlagen der Informatik Algorithmen und Datenstrukturen Objektorientierte Programmierung Grundlagen und Konzepte Englisch	
3 Software Engineering - Analyse Statistik Objektorientierte Programmierung Vertiefung Datenbanken Rechnernetze Betriebssysteme	
4 Numerische Mathematik Software Engineering - Design Echtzeitsysteme Datenschutz / IT Security IT-Projekte	
5 Studium Generale Aktuelle Themen der Informatik Informatik-Projekt Verteilte Anwendungen Moderne Netzstrukturen	
6 Praxisphase Bachelorarbeit mit Kolloquium	 Alle Module der ersten 4 Studiensemester Nachweis des Beginns der Praxisphase in Form eines unterzeichneten Ausbildungsvertrages sowie erfolgreicher Abschluss aller Module der ersten 5 Studiensemester
Praxisphase betreutes Praxisprojekt	Anfang zwischen 1.3 und 1.4 Dauer 14 Wochen a 4,5Tage
Praxisphasen-Seminar	Jeden Freitag vormittags Seminar an der Fachhochschule 18 Teilnehmer je 20 Min. Vortrag und 10 Min. Diskussion
Zulassungsvoraussetzung	alle Module der ersten 4 Semester



Internationalisierung

Zur Förderung der Attraktivität des Studienangebots für ausländische Studierende ist vorgesehen, einen der Züge des dritten und vierten Studienseesters komplett in englischer Sprache durchzuführen. Auf Antrag können in englischsprachlichen Modulen der dritten und vierten Semesters die Prüfungen auch in deutscher Sprache durchgeführt werden. Des weiteren können in allen anderen Modulen die Leistungsnachweise nach Genehmigung durch den Prüfungsausschuss auch in englischer Sprache durchgeführt werden.

Um die Mobilität der inländischen Studierenden zu fördern, kann auf Antrag die Praxisphase mit anschließender Bachelorarbeit im Ausland durchgeführt werden.



Module des ersten Studienjahrs

1. Modul: Mathematische Grundlagen – Algebra	
Code	
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur über 90 Minuten Auf Antrag kann die Prüfung auch in englischer Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Basic level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Das Modul behandelt Grundlagen der Algebra, die für das Verständnis der strukturellen und logischen Grundlagen elektronischer Informationsverarbeitung und die Erarbeitung von Lösungswegen wesentlich sind. Vorausgesetzt werden die zum Erwerb der Fachhochschulreife erwarteten Kenntnisse der Mathematik.
Lernergebnis / Kompetenzen	In dem Modul werden strukturelle und logische Grundlagen elektronischer Informationsverarbeitung vermittelt. Im Vordergrund steht dabei die sichere Handhabung von abstrakten Begriffen und Methoden. Die abstrakte Umsetzung von konkreten Problemen der Anwendung in formale Modelle wird hier gefördert. Das Modul trägt somit zum Ausbau der Methodenkompetenz bei und fördert die Qualifikation zum Umgang mit abstrakten Methoden, Strukturen und Mustern. Neben der Vermittlung der Inhalte ist es ausdrückliches Ziel, die Studierenden mit dem Umgang mit abstrakten mathematischen Begriffen vertraut zu machen. Die Studierenden sollen in der Lage sein, sich abstrakte Begriffe selbständig erarbeiten und grundlegende Techniken oder Verfahren aneignen zu können. Die Studierenden sollen die mathematischen Grundlagenbegriffe kennen, sie verstanden haben und dazu in der Lage sein, sie auch in anderen Zusammenhängen anwenden zu können. Sie sollen die wichtigsten Begriffe, Strukturen und Methoden der elementaren Algebra und der linearen Algebra kennen. Insbesondere sollen sie mit den algebraischen Grundstrukturen, die für das Verständnis formaler Strukturen der Informatik notwendig sind, gut vertraut sein und den Umgang mit ihnen sicher beherrschen.
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Mathematische Grundlagen – Algebra, Übung Mathematische Grundlagen – Algebra
Häufigkeit des Angebots	nur im Wintersemester
Anerkannte Module	Keine
Modulkoordination	Falkenberg



Hinweise	
----------	--

1.1 Unit: Vorlesung Mathematische Grundlagen – Algebra	
Code	
Modulname	Mathematische Grundlagen – Algebra
Lernform	Vorlesung 4 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	100 h
Anteil Präsenzzeit	70 h
Anteil Selbststudium	30 h
Inhalte	<p>Die Unit behandelt Grundlagen der Algebra, die für das Verständnis der strukturellen und logischen Grundlagen elektronischer Informationsverarbeitung und die Erarbeitung von Lösungswegen wesentlich sind. Daher bauen alle weiterführenden Mathematik-Module (Diskrete Mathematik, Statistik) auf diesem auf. Ferner ist dieses Modul eine der Grundlagen der Informatik-Module ab dem zweiten Semester, bei denen formale Modelle und Methoden im Vordergrund stehen.</p> <p>Das Modul widmet sich drei größeren Themenblöcken:</p> <ul style="list-style-type: none"> • Grundlagen: <ul style="list-style-type: none"> o Aussagenlogik und Beweistechniken o Mengen und ihre Darstellung o Relationen, insbes. Ordnungs- und Äquivalenzrelationen o Vollständige Induktion und Rekursion o Halbgruppen und Gruppen • Lineare Algebra <ul style="list-style-type: none"> o Vektorräume und verwandte algebraische Strukturen o Lineare Gleichungssysteme, Gauß-Verfahren etc. o Lineare Abhängigkeit und Unabhängigkeit, Basis, Dimension o Matrizen, insbes. algebraische Operationen mit Matrizen o Lineare Abbildungen • Elementare Zahlentheorie <ul style="list-style-type: none"> o Rechnen in ganzen Zahlen, insbes. Primzahlen, ggT, kgV, euklidischer Algorithmus o Modulare Arithmetik, Restklassen, lineare Kongruenzen • Einführung in kryptografische Verfahren
Literatur	<ul style="list-style-type: none"> • Brill M: Mathematik für Informatiker, Hanser • Dörfler W., Peschek W.: Einführung in die Mathematik für Informatiker, Hanser • Fetzer/Fränkell: Mathematik für Fachhochschulen 1 -3 , VDI • Gelrich/Gelrich: Mathematik , Harri Deutsch • Hartmann P.: Mathematik für Informatiker, Vieweg



	<ul style="list-style-type: none"> • Meinel/Mundhenk, Mathematische Grundlagen der Informatik, Teubner • Papula, Mathematik für Ingenieure 1 - 3 , Vieweg • Preuß/Wenisch, Lehr- und Übungsbuch Mathematik, Fachbuchverlag Leipzig • Schaums Überblicke, McGraw Hill • Stigl, Mathematik für Fachhochschulen, Hanser Verlag Witt K.-U.: Algebraische Grundlagen der Informatik, Vieweg
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Birkholz, Falkenberg, Hackenbracht, Schubert, Selder
Hinweise	

1.2 Unit: Übung Mathematische Grundlagen – Algebra	
Code	
Modulname	Mathematische Grundlagen – Algebra
Lernform	Übung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	50 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	15 h
Inhalte	Aufgaben und Beispiele zu den Themen des Moduls Die Übungen dienen dazu, dass die Studierenden lernen, eine die algebraischen Fragestellungen der Aufgabe zu verstehen und mit den behandelten Methoden zu lösen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Mathematische Grundlagen – Algebra
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Birkholz, Falkenberg, Hackenbracht, Schubert, Selder
Hinweise	

2. Modul: Grundlagen der Informatik und maschinennahe Programmierung	
Code	
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	10 CP
Prüfungsart	Klausur am Rechner über 120 Minuten Auf Antrag kann die Prüfung auch in englischer Sprache stattfinden.



Status	Pflichtmodul
Niveaustufe / Level	Basic level course
Notwendige Voraussetzungen	Erfolgreicher Abschluss der Unit Übung maschinennahe Programmierung
Empfohlene Voraussetzungen	Das Modul liefert als Einführung in die Informatik allgemeine und grundlegende Kenntnisse und Fähigkeiten für das gesamte Studium und setzt daher keine Kenntnisse der Informatik voraus.
Lernergebnis / Kompetenzen	<p>Durch dieses Modul sollen wichtige Informationen zur Hardware des Computers und zum Zusammenwirken seiner Komponenten, durch die direkte Programmierung des Prozessors ein Verständnis für die Hardware als Grundlage allen Programmierens erzeugt werden.</p> <p>Nach erfolgreichem Abschluss des Moduls sollen die Studierenden</p> <ul style="list-style-type: none"> • das klassische Rechnermodell nach von Neumann am Beispiel des aktuellen Personal Computers erläutern können, • die Verarbeitung von Zahlen auf der Hardware-Ebene verstehen und einfache Logikbausteine und ihre Verwendung an einfachen Beispielen erläutern können, • den Aufbau und die Wirkungsweise eines Mikroprozessors am Beispiel des Intelprozessors Pentium und Nachfolger kennen, • Assemblerprogramme erstellen und lauffähig machen können, die die wichtigsten Prozessorbefehle verwenden, • die Vorgänge im Rechner bei Programmerstellung und Programmabläufen und den Zusammenhang von maschinen- und problemorientierter Programmierung verstehen.
Arbeitsaufwand (h) / Gesamtworkload	300 h
Units	Vorlesung Grundlagen der Informatik und maschinennahe Programmierung, Übung maschinennahe Programmierung
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	Keine
Modulkoordination	Wolf
Hinweise	

2.1 Unit: Vorlesung Grundlagen der Informatik und maschinennahe Programmierung

Code	
Modulname	Grundlagen der Informatik und maschinennahe Programmierung
Lernform	Vorlesung 4 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	150
Anteil Präsenzzeit	70 h
Anteil Selbststudium	80 h



<p>Inhalte</p>	<p>Das Modul liefert als Einführung in die Informatik allgemeine und grundlegende Kenntnisse und Fähigkeiten für das gesamte Studium, insbesondere für die Module, die sich mit Computer-Hardware beschäftigen. Inhalte des Moduls werden im Modul 7: Digitaltechnik in detaillierter Betrachtung weitergeführt. Im Zentrum der Vorlesung steht die Befassung mit dem Aufbau und der hardwarenah beschriebenen Wirkungsweise eines Rechners und der Assemblerprogrammierung.</p> <ul style="list-style-type: none"> • Informationsdarstellung im Rechner (Speicherformate, Logische Größen, Zeichen, Zahlen, Zahlensysteme, Ganze Zahlen, Gleitkommazahlen) • von Neumann-Rechnerarchitektur (Kurzer Abriss der Rechnerentwicklung, von-Neumann-Modell, Nicht-von-Neumann-Rechner, Arbeitsweise des Prozessors, Aufbau und Zusammenwirken von Prozessor und Speicher) • Verarbeitung von Zahlen (Ein- und Ausgabe, Ganzzahlen-Arithmetik, Kontrolle der Arithmetik, Gleitkomma-Arithmetik, Genauigkeit) • Logik-/Arithmetikbausteine (Basisbausteine und Kombinationen, Halb- und Volladdierer) • Betriebssysteme (Überblick, Aufgaben, Klassifikation, Hauptkomponenten) • Massenspeicher (Festplatte, Optische Speicher) • Allgemeiner Aufbau eines Assemblerprogramms • Assemblersprachelemente, Makros • Werkzeuge der Assemblerprogrammierung • Adressierungsarten (Register, Arbeitsspeicher, Stack) • Befehlsgruppen • Programmsteuerung, Labels und Flags • Ein- und Ausgabe von ganzen Zahlen • Unterprogramme, Parameterübergabe, Bibliotheken • Dateibehandlung • Bitoperationen • Verbindung von Assembler- und Hochsprachenprogrammen, gegenseitiger Aufruf, Vergleich mit Compiler generiertem Code • Programmierung des mathematischen Coprozessors
<p>Literatur</p>	<p>Grundlagen der Informatik</p> <ul style="list-style-type: none"> • Appelrath H.-J., Ludwig J., Stuttgart: Skriptum Informatik – eine konventionelle Einführung, B. G. Teubner Verlag 1991 • Rembold U., München: Einführung in die Informatik für Naturwissenschaftler und Ingenieure, Carl Hanser Verlag 1987 • Blieberger J., Schildt G.-H., Schmid U., Stöckler S., Wien, New York:



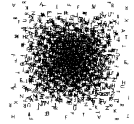
	<p>Informatik, Springer Verlag 1990</p> <ul style="list-style-type: none"> • Breuer H., München: dtv-Atlas zur Informatik, Tafeln und Texte (Band 2490) Deutscher Taschenbuch Verlag • Messmer H.-P., Bonn [u.a.]: PC-Hardwarebuch, Addison-Wesley-Longman 1998 • Schneider U., Werner D., München Wien: Taschenbuch der Informatik, Fachbuchverlag Leipzig im Carl Hanser Verlag 2000 • H.P. Gumm, M. Sommer: Einführung in die Informatik. Addison Wesley, Bonn, neueste Auflage • G. Küveler, D. Schwoch: Arbeitsbuch Informatik. Vieweg, Braunschweig, 1996 • W. Merzenich, H.C. Zeidler: Informatik für Ingenieure. Teubner, Stuttgart, 1997 • R. Richter, P. Sander, W. Stucky: Der Rechner als System. Teubner, Stuttgart, 1997 • Rechenberg, P., Was ist Informatik, Hanser, 2000 • Lutz, M., Schmitt, F., Vom Prozessor zum Programm, Fachbuchverlag Leipzig, 1997 <p>Maschinennahe Programmierung: Der in der Lehrveranstaltung benutzte GNU-Assembler verwendet die sogenannte AT&T-Notation, wohingegen die im Buchhandel erhältliche Literatur ausschließlich die Intel-Notation verwendet. In den Literaturhinweisen ist der Übergang zwischen den beiden Notationen erläutert.</p> <ul style="list-style-type: none"> • Elsner, D., Fenlason, J. & friends: The GNU Assembler, 1994. Im Internet verfügbar unter: http://www.gnu.org/manual/gas-2.9.1/html_mono/as.html/ (Stand August 2001) • Online-Handbücher und Online-Hilfen unter Linux • Bücher, die eine Auflistung der Assemblerbefehle des Intelprozessors 80x86/Pentium enthalten, z. B. Wohak, B., Maurus, R.: 80x86/Pentium • Assembler Programmierung unter DOS und Windows. Bonn; Albany: IWT Verlag GmbH 1995 • Dumbacher, B.: Die Assemblerprogrammierung der Prozessoren 80x86/Pentium, Manuskript zur dieser Lehrveranstaltung 2001
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Nauth, Pitka, Wolf
Hinweise	

2.2 Unit: Übung maschinennahe Programmierung	
Code	
Modulname	Grundlagen der Informatik und maschinennahe Programmierung
Lernform	Übung 2 SWS



Sprache	Deutsch
Arbeitsaufwand (h) / workload	150 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	115 h
Inhalte	In den Übungen zur maschinennahen Programmierung werden die Inhalte der Vorlesung durch praktische Tätigkeit am Rechner angewandt. Die Übungen dienen dazu, dass die Studierenden lernen, eine Aufgabe zu verstehen und zu auf dem Rechner mittels eines maschinennahen Programms umzusetzen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Grundlagen der Informatik und maschinennahe Programmierung
Art und Form des Leistungsnachweises	Übungen am Rechner; Teilnahme an mindestens 80% aller Übungen ist Zulassungsvoraussetzung für die Modulprüfungsleistung
Bewertung	undifferenziert
Lehrende	Nauth, Pitka, Wolf
Hinweise	

3. Modul: Einführung in die Programmierung mit C	
Code	
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	10 CP
Prüfungsart	Klausur am Rechner über 120 Minuten Auf Antrag kann die Prüfung auch in englischer Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Basic level course
Notwendige Voraussetzungen	Erfolgreicher Abschluss der Units Übung Programmierung in C und Unix Praktikum
Empfohlene Voraussetzungen	Das Modul stellt den Einstieg in das Programmieren dar und setzt keine Kenntnisse der Programmierung voraus.
Lernergebnis / Kompetenzen	Das Modul liefert die Grundkenntnisse der prozeduralen Programmierung und praktische Fähigkeiten im Umgang mit dem Betriebssystem Unix für alle weiteren Veranstaltungen, welche Programmierung als Schwerpunkt haben. Es legt damit den Grundstein wesentlicher Fähigkeiten des Informatik-Absolventen. Nach erfolgreichem Abschluss des Moduls sollen die Studierenden <ul style="list-style-type: none"> • Lösungen für einfache Aufgabenstellungen als Struktogramm formulieren und Struktogramme in eine höheren Programmiersprache (derzeit in C) umsetzen können, • Methoden zur Fehlererkennung und Fehlerbeseitigung beherrschen, • die wichtigsten Sprachelemente und Bibliotheksfunktionen verwenden



	können, <ul style="list-style-type: none"> • Grundprinzipien und Aufgaben von Betriebssystemen verstehen und am Beispiel des Betriebssystems UNIX/LINUX die wichtigsten Kommandos, Werkzeuge und einfache Skripts beherrschen.
Arbeitsaufwand (h) / Gesamtworkload	300 h
Units	Vorlesung Einführung in die Programmierung mit C, Übung Programmierung mit C, UNIX Praktikum
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	keine
Modulkoordination	N. N.
Hinweise	

3.1 Unit: Vorlesung Einführung in die Programmierung mit C

Code	
Modulname	Einführung in die Programmierung mit C
Lernform	Vorlesung 2 SWS
Sprache	Deutsch oder Englisch
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	<p>Das Modul stellt den Einstieg in das Programmieren dar und ist damit direkte Grundlage für die folgenden Module der Programmierung und darüber hinaus aber wichtige Basis für fast alle Module des Studiengangs.</p> <ul style="list-style-type: none"> • Einführendes Beispiel mit grundlegenden Sprachelementen • Analyse einfacher Aufgabenstellungen und Erstellen eines strukturierten Lösungsvorschlags • Editieren, Übersetzen, Ausführen von Programmen • Programmentwurf • Elementare Datentypen, Variablen und Arithmetik • Ein-/Ausgabe • Verzweigung und Schleifen • Felder, Zeichenkette • Fehlersuche und Fehlerbeseitigung, Verwendung von Werkzeugen • Zeiger, dynamische Speicherverwaltung • Unterprogramme (Funktionen) und Parameter, modularer Programmaufbau, Bibliotheksfunktionen • Dateien • Strukturierte Datentypen
Literatur	<ul style="list-style-type: none"> • Kernighan, B., W., Ritchie, D., M., Programmieren in C, Hanser, 1990



	<ul style="list-style-type: none"> • RRZN, "C" • Peter A. Darnell, Philip E. Margolis: C, A Software Engineering Approach • Herold, Helmut; Arndt, Jörg. C-Programmierung unter Linux. 2002 SuSE Press, Nürnberg • Mittelbach, H., Einführung in C, Fachbuchverlag Leipzig, 2001 • Erlenkötter, H., C Programmieren von Anfang an, Rowohlt, 1999 • Schneider, U., Werner, D., Taschenbuch der Informatik, Fachbuchverlag Leipzig, 2001
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Alle Informatik Dozenten
Hinweise	

3.2 Unit: Übung Programmierung mit C

Code	
Modulname	Einführung in die Programmierung mit C
Lernform	Übung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	120 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	85 h
Inhalte	In den Übungen zur Einführung in die Programmierung mit C werden die Inhalte des Moduls durch praktische Tätigkeit am Rechner angewandt. Die Übungen dienen dazu, dass die Studierenden lernen, eine Aufgabe zu verstehen und zu auf dem Rechner mittels eines C Programms umzusetzen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Einführung in die Programmierung mit C
Art und Form des Leistungsnachweises	Übungen am Rechner; Teilnahme an mindestens 80% aller Übungen ist Zulassungsvoraussetzung für die Modulprüfungsleistung
Bewertung	Undifferenziert
Lehrende	Alle Informatik-Dozenten
Hinweise	

3.3 Unit: UNIX Praktikum

Code	
Modulname	Einführung in die Programmierung mit C
Lernform	Übung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	95 h



Anteil Präsenzzeit	35 h
Anteil Selbststudium	60 h
Inhalte	<p>Das Praktikum dient zur Einführung in den Umgang mit dem Rechner auf Betriebssystemebene. Ziel ist es eine Basis zu legen, die einen gezielten Einsatz von UNIX-artigen Betriebssystemen ermöglicht. Hierzu sollen Fähigkeiten und Fertigkeiten in der Benutzung von und der Programmierung unter UNIX-artigen Betriebssystemen entwickelt werden. Am Ende der Lehrveranstaltung sollen die Studierenden die Grundoperationen des Betriebssystems UNIX beherrschen und sie sollen in der Lage sein, einfache Shellskripte zu schreiben. Durch die kontinuierliche Erfolgskontrolle bekommen die Studierenden ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.</p> <p>Neben einem Überblick über das UNIX-Betriebssystem wird auf die Benutzung eingegangen. Zudem wird auf die Programmierung unter UNIX vorbereitet. Im Einzelnen werden folgenden Themen behandelt:</p> <ul style="list-style-type: none"> • Einstieg, Allgemeines, Dateien, Prozesse • Bildschirmorientierte Oberflächen • Interaktive Kommandointerpretation • Texteditoren • Dateikommandos • Einfache Shellskripte
Literatur	<ul style="list-style-type: none"> • RRZN, Hannover. Einführung in das Arbeiten unter Unix. • Herold, Unix-Grundlagen, Addison-Wesley, 2.Auflage, 1992 • Herold, Unix-Shells, Addison-Wesley, 1.Auflage, 1992
Art und Form des Leistungsnachweises	Übungen am Rechner; Teilnahme an mindestens 80% aller Übungen ist Zulassungsvoraussetzung für die Modulprüfungsleistung
Bewertung	Undifferenziert
Lehrende	Falkenberg, GÜsmann, Hannemann, Selder
Hinweise	

4. Modul: Betriebswirtschaftslehre	
Code	
Studiengang	Informatik
Verwendbarkeit	verwendbar in anderen Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur über 90 Minuten Auf Antrag kann die Prüfung auch in englischer Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Basic level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Keine



Lernergebnis / Kompetenzen	Das Modul behandelt die Grundlagen der Allgemeinen Betriebswirtschaftslehre. Der Studierende soll in die Lage versetzt werden, von den betrieblichen Funktionsbereichen die Verbindung zur informationstechnologischen Unterstützung im Betrieb und zwischenbetrieblich herzustellen und zu verstehen. Das Modul trägt damit zur Förderung des Einblicks in wichtige Anwendungsfelder der Informatik und des Verständnis der Bedeutung der IT für das Unternehmen und die Gesellschaft bei. Die Studierenden sollen die Grundbegriffe des Wirtschaftens, der Organisation, des Rechnungswesens, des Personalwesens, des Investitions- und Finanzierungsbereiches sowie die betrieblichen Funktionsbereiche der Materialwirtschaft, der Produktion und
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Betriebswirtschaftslehre
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	Keine
Modulkoordination	N. N.
Hinweise	

4.1 Unit: Vorlesung Betriebswirtschaftslehre	
Code	
Modulname	Betriebswirtschaftslehre
Lernform	Vorlesung 4 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	150 h
Anteil Präsenzzeit	70 h
Anteil Selbststudium	80 h
Inhalte	Im Rahmen der Lehrveranstaltung werden behandelt: <ul style="list-style-type: none"> • Wirtschaft, Betrieb, Unternehmen und BWL • Betriebliche Organisation • Rechnungswesen und Steuerung in Betrieb und Unternehmen • Marketing Personalwirtschaft und Produktion • IT und Business
Literatur	<ul style="list-style-type: none"> • Wöhe: Einführung in die Allgemeine BWL, 21. Auflage, München 2002 • Ott: Betriebswirtschaftslehre für Ingenieure und Informatiker, München 1995 • Schierenbeck: Grundzüge der Betriebswirtschaftslehre, 16. Auflage 2002, München • Tommen/Achleitner: Allg. Betriebswirtschaftslehre, 4. Auflage 2003, Wiesbaden • Homburg: Quantitative Betriebswirtschaftslehre, 3. Auflage 2000, Wies-



	Wiesbaden
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	N.N. (Fachdozent des Fachbereich 3)
Hinweise	

5. Modul: Englisch	
Code	
Studiengang	Informatik
Verwendbarkeit	verwendbar in anderen Studiengängen
Dauer	2 Semester
Credits	5 CP
Prüfungsart	Klausur/mündliche Prüfung bzw. Referate, Essays / schriftliche Hausarbeiten Die Modulprüfungsleistung wird im Rahmen der Unit Englisch 2 erbracht.
Status	Pflichtmodul
Niveaustufe / Level	Intermediate level course
Notwendige Voraussetzungen	Erfolgreicher Abschluss der Unit Englisch 1
Empfohlene Voraussetzungen	Solide alltagspraktische Vorkenntnisse – mind. 6 Jahre Schulunterricht – oder Besuch eines Vorbereitungskurses
Lernergebnis / Kompetenzen	Ziel des Moduls ist, die Kommunikationsfähigkeit in englischer Sprache zu verbessern. Dabei soll die Bewältigung berufstypischer Situationen, das heißt auch das Erlernen des fachspezifischen Vokabulars, im Vordergrund stehen. Die Kommunikationsfähigkeit der Studenten in englischer Sprache soll dergestalt verbessert werden, dass sie <ul style="list-style-type: none"> • den allgemeinen Anforderungen, an eine berufliche Kommunikation in Englisch gewachsen sind und • berufstypische Situationen in der Projektarbeit innerhalb international zusammengesetzter Teams bei mittleren und größeren Unternehmen bewältigen können. Das Modul trägt durch die Förderung und den Ausbau der Sprachkompetenz in Englisch zum Erwerb der außerfachlichen Kompetenzen bei.
Arbeitsaufwand (h) / Gesamtworkload	150 h anteiliger Workload zum Erwerb außerfachlicher Kompetenzen
Units	Englisch 1, Englisch 2
Häufigkeit des Angebots	Das Modul beginnt jeweils nur im Wintersemester.
Anerkannte Module	Keine
Modulkoordination	Lindemann
Hinweise	



5.1 Unit: Englisch 1	
Code	
Modulname	Englisch 1
Lernform	Übung 2 SWS
Sprache	Englisch
Arbeitsaufwand (h) / workload	60 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	25 h
Inhalte	<p>Erarbeiten (Auffrischen) der notwendigen grammatikalischen Grundkenntnisse sowie des erforderlichen Vokabulars. Typische berufsbezogene Kommunikationssituationen wie</p> <ul style="list-style-type: none"> • Meeting Visitors • Office Communication (Telephoning, Letter Writing) • Presenting in Englisch <p>werden eingeübt.</p>
Literatur	----
Art und Form des Leistungsnachweises	Teilnahme an mindestens 80% aller Übungen ist Zulassungsvoraussetzung für die Modulprüfungsleistung
Bewertung	Undifferenziert
Lehrende	Sprachdozenten des Fachbereich 3
Hinweise	

5.2 Unit: Englisch 2	
Code	
Modulname	Englisch
Lernform	Übung 3 SWS
Sprache	Englisch
Arbeitsaufwand (h) / workload	90 h
Anteil Präsenzzeit	50 h
Anteil Selbststudium	40 h
Inhalte	<p>Erarbeiten (Auffrischen) der notwendigen grammatikalischen Grundkenntnisse sowie des erforderlichen Vokabulars. Typische berufsbezogene Kommunikationssituationen wie</p> <ul style="list-style-type: none"> • Meeting Visitors • Office Communication (Telephoning, Letter Writing) • Presenting in Englisch <p>werden eingeübt.</p>
Literatur	----
Art und Form des Leistungsnachweises	Klausur/mündliche Prüfung bzw. Referate, Essays / schriftliche Hausarbeiten
Bewertung	differenziert
Lehrende	Sprachdozenten des Fachbereich 3



Hinweise	
----------	--

6. Modul: Diskrete Mathematik	
Code	
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur über 90 Minuten Auf Antrag kann die Prüfung auch in englischer Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Intermediate level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Inhaltlich wird in dem Modul das Modul Mathematische Grundlagen des ersten Studienseesters vorausgesetzt.
Lernergebnis / Kompetenzen	<p>Aufbauend auf dem Modul Grundlagen der Mathematik soll mit dem Modul die Fähigkeit, mit abstrakten Begriffen zu operieren, vertieft werden. Dies stellt eine Schlüsselqualifikation für die Informatik dar. Konkret sollen mit den Modul die wichtigsten mathematischen Techniken für Anwendungen in den Kerndisziplinen der Informatik (Theoretische Informatik und Rechnerarchitektur, Betriebssysteme, Rechner netze etc...) vermittelt werden. Nach erfolgreichem Abschluss des Moduls sollen die Studierenden</p> <ul style="list-style-type: none"> • die erlernten Techniken anwenden können. • die behandelten Begriffe und Algorithmen verstanden haben, sie geeignet modifizieren können und ggf. neue Lösungsmöglichkeiten zu verwandten Problemen entwickeln können. • in der Lage sein, mathematische Begriffe aus dem Umfeld der behandelten Themen sich selbständig aneignen zu können, diese Begriffe in größere Zusammenhänge einbetten zu können und relevante Verfahren und Techniken für Probleme im Umfeld dieser Begriffe entwickeln zu können. • die wichtigsten mathematischen Begriff, Verfahren und Techniken soweit haben, dass sie in den Folgekursen der Informatik in konkreten Problemstellungen adäquat angewendet werden können.
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Diskrete Mathematik, Übung Diskrete Mathematik
Häufigkeit des Angebots	Nur im Sommersemester
Anerkannte Module	Keine
Modulkoordination	Falkenberg
Hinweise	



6.1 Unit: Vorlesung Diskrete Mathematik	
Code	
Modulname	Diskrete Mathematik
Lernform	Vorlesung 4 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	100 h
Anteil Präsenzzeit	70 h
Anteil Selbststudium	30 h
Inhalte	<p>In der Vorlesung sollen Inhalte vermittelt werden, die für spätere Mathematik- und Informatikmodule erforderlich sind.</p> <ul style="list-style-type: none"> • Zählkoeffizienten, Zählprinzipien, Abzähltechniken (Rekursion, ...) • Einführung in die Codierungstheorie • Boolesche Algebra (boolesche Terme und Funktionen, Normalformen, Minimierungsalgorithmen ...) • Graphentheorie (Bäume, kürzeste Wege, bipartite Graphen, Matchings ...) <p>In den parallel stattfindenden Modulen Algorithmen und Datenstrukturen, Objektorientierte Programmierung mit C++ werden die Inhalte unter verschiedenen Blickwinkel der Informatik betrachtet und angewendet. Das Modul liefert inhaltliche Voraussetzungen für die Folgemodule Theoretische Grundlagen der Informatik (algebraische Strukturen, boolesche Algebra), Rechnernetze, Betriebssysteme (Graphentheorie), Statistik (Abzähltechniken), Datenschutz / IT-Security (Codierung, modulare Arithmetik, Zahlentheorie, Polynomringe)</p>
Literatur	<ul style="list-style-type: none"> • Martin Aigner: Diskrete Mathematik, Vieweg Verlag, 1999 • R. Bodensiek, R. Lang; Lehrbuch der Graphentheorie, Band 1 und 2, Spektrum Akademischer Verlag, 1995 • R. Diestel: Graphentheorie, Springer Verlag, 1996 • R. P. Grimaldi: Discrete and Combinatorial Mathematics, Addison Wesley, 1999 • P. Gritzmann, R. Brandenburg: Das Geheimnis des kürzesten Weges, Springer Verlag, 2002 • Thomas Ihringer: Diskrete Mathematik, Teubner Verlag, 1994 • K. Neumann, M. Morlock: Operations Research, Hanser 1993 • T. Rauber: Algorithmen in der Computergraphik, Teubner Verlag, 1993 • A. Steger: Diskrete Strukturen 1, Springer Verlag 2001 • Peter Stingl: Mathematik für Fachhochschulen, Kapitel 2 und 6, Hanser Verlag, 1996 • J. Truss: Discrete Mathematics for Computer Scientists, Addison Wesley, 1999 • Wayne L., Winston: Operations Research, Applications and Algorithms, Winston, 1991



	<ul style="list-style-type: none"> • K.-U. Witt: Algebraische Grundlagen der Informatik, Vieweg Verlag 2000 • Arithmeum - Old Problems in Discrete Mathematics and its Modern Applications, Springer Verlag, 2002
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Birkholz, Falkenberg, Schubert, Selder
Hinweise	

6.2 Unit: Übung Diskrete Mathematik

Code	
Modulname	Diskrete Mathematik
Lernform	Übung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	50 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	15 h
Inhalte	Aufgaben und Beispiele zu den Themen des Moduls Die Übungen dienen dazu, dass die Studierenden lernen, eine die Aufgabe zu verstehen und mit den behandelten Methoden zu lösen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Diskrete Mathematik
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Birkholz, Falkenberg, Schubert, Selder
Hinweise	

7. Modul: Digitaltechnik

Code	
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur über 90 Minuten Auf Antrag kann die Prüfung auch in englischer Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Basic level course
Notwendige Voraussetzungen	Erfolgreicher Abschluss der Unit Übung Digitaltechnik
Empfohlene Voraus-	Keine



setzungen	
Lernergebnis / Kompetenzen	<p>Die Digitaltechnik bildet die Hardwarebasis für die Informatik. Viele Mechanismen der Datenverarbeitung lassen sich in Hardware und in Software realisieren. Die Anweisungen in Maschinenprogrammen beziehen sich direkt auf Hardwareeinheiten. In diesem Sinne legt die Lehrveranstaltung Digitaltechnik die Hardwarebasis für Lehrveranstaltungen in der Informatik. Lernziele des Moduls sind</p> <ul style="list-style-type: none"> • Kenntnisse elektrotechnischer Randbedingungen beim Betrieb digitaler Schaltungen • Kenntnisse des systematischen Entwurfs digitaler Schaltungen • Fähigkeit zur Lösung grundlegender Aufgaben.
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Digitaltechnik, Übung Digitaltechnik
Häufigkeit des Angebots	Nur im Sommersemester
Anerkannte Module	Keine
Modulkoordination	Pitka
Hinweise	

7.1 Unit: Vorlesung Digitaltechnik	
Code	
Modulname	Digitaltechnik
Lernform	Vorlesung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	65 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	<ul style="list-style-type: none"> • Schaltnetze • Speicherglieder • Schaltwerke • Physikalische Grundlagen • Halbleiterbauteile • Elektronische Elemente
Literatur	<ul style="list-style-type: none"> • Schiffmann / Schmitz: Technische Informatik 1 • Borgmeyer: Grundlagen der Digitaltechnik, Hanser-Verlag
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Frömel, Pitka
Hinweise	



7.2 Unit: Übung Digitaltechnik	
Code	
Modulname	Digitaltechnik
Lernform	Übung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	Die theoretischen und praktischen Übungen dienen dazu, dass die Studierenden die in der Vorlesung vermittelten theoretischen Inhalte anhand von Aufgaben zu verstehen, zu lösen und praktisch im Labor umzusetzen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Digitaltechnik
Art und Form des Leistungsnachweises	Teilnahme an mindestens 80% aller Übungen ist Zulassungsvoraussetzung für die Modulprüfungsleistung
Bewertung	Keine
Lehrende	Frömel, Pitka
Hinweise	

8. Modul: Theoretische Grundlagen der Informatik	
Code	
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5
Prüfungsart	Klausur über die Inhalte der Units Automatentheorie und formale Sprachen und Rechnerarchitektur im Umfang von 120 Minuten Auf Antrag kann die Prüfung auch in englischer Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Basic level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Das Modul steht in enger Beziehung zu den parallel stattfindenden Modulen Diskrete Mathematik und Digitaltechnik. Es setzt die Inhalte des Moduls Mathematische Grundlagen voraus.
Lernergebnis / Kompetenzen	In dem Modul wird einerseits Grundlagenwissen über Programmiersprachen und andererseits grundlegendes Wissen über die Arbeitsweise von Computern vermittelt. Die Studierenden sollen die grundlegenden Konzepte von Automaten und formalen Sprachen kennenlernen und deren Bedeutung für höhere Programmiersprachen und andere Bereiche der Informatik verstanden haben. Ferner sollen sie die wichtigsten architektonischen Prinzipien und Ebenen für den Entwurf von Rechenanlagen verstanden haben. Im parallel stattfindenden Modul Digitaltechnik wird die unterste Ebene der Logik bzw. Hardware ausführlich behandelt.



Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Rechnerarchitektur, Übung Rechnerarchitektur, Vorlesung Automatentheorie und formale Sprachen, Übung Automatentheorie und formale Sprachen
Häufigkeit des Angebots	Nur im Sommersemester
Anerkannte Module	Keine
Modulkoordination	N. N.
Hinweise	

8.1 Unit: Vorlesung Rechnerarchitektur

Code	
Modulname	Theoretische Grundlagen der Informatik
Lernform	Vorlesung 1 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	32 h
Anteil Präsenzzeit	17 h
Anteil Selbststudium	15 h
Inhalte	<p>Nach einer Einführung wird der Aufbau von Computer-Systemen behandelt. Anschließend werden die verschiedenen Architekturebenen</p> <ul style="list-style-type: none"> • Digitale logische Ebene • Mikroarchitekturebene • ISA-Ebene (Instruction Set Architecture) • Ebene der Betriebssystemmaschine • Ebene der Assemblersprache <p>betrachtet. Am Ende der Lehrveranstaltung werden alternative Rechnerarchitekturen betrachtet.</p>
Literatur	<ul style="list-style-type: none"> • Carpinelle, John D.: Computer Systemes , Organization & Architecture, Addison Wesley 2001 • Clements, Alan: The Principles of Computer Hardware; Oxford University Press, 3. Auflage 2000 • Coy, Wolfgang: Aufbau und Arbeitsweise von Rechenanlagen, Vieweg, neueste Auflage • Gumm, Heiz-Peter; Sommer, Manfred: Einführung in die Informatik, Oldenbourg Verlag, München, vierte Auflage • Hennesy, John L.; Patterson, David A.: Rechnerarchitektur, Vieweg 1994 • Oberschelp, Walter; Vossen, Gottfried: Rechneraufbau und Rechnerstrukturen, Oldenbourg Verlag München 1997 • Tannenbaum, Andrew S.; Godman, James: Computerarchitektur, Prentice Hall, 4. Auflage



Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Doeben-Henisch, Filip, Rauch, Selder.
Hinweise	

8.2 Unit: Übung Rechnerarchitektur

Code	
Modulname	Theoretische Grundlagen der Informatik
Lernform	Übung 1 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	43 h
Anteil Präsenzzeit	17 h
Anteil Selbststudium	26 h
Inhalte	Aufgaben und Beispiele zu den Themen der Unit Vorlesung Rechnerarchitektur Die Übungen dienen dazu, dass die Studierenden lernen, eine die algebraischen Fragestellungen der Aufgabe zu verstehen und mit den behandelten Methoden zu lösen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Rechnerarchitektur
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Doeben-Henisch, Filip, Rauch, Selder.
Hinweise	

8.3 Unit: Vorlesung Automatentheorie und formale Sprachen

Code	
Modulname	Theoretische Grundlagen der Informatik
Lernform	Vorlesung 1 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	32 h
Anteil Präsenzzeit	17 h
Anteil Selbststudium	15 h
Inhalte	In der Vorlesung Automatentheorie und formale Sprachen werden theoretische Grundlagen vermittelt auf die in vielen Informatikmodulen aufgebaut wird. <ul style="list-style-type: none"> • Formale Sprachen und ihre Klassifizierung in der Chomsky-Hierarchie. • Endliche Automaten, Kellerautomaten, Turingmaschinen und ihr Zusammenhang mit formalen Sprachen • Anfänge der Theorie der Berechenbarkeit und Komplexität



Literatur	<ul style="list-style-type: none"> • E. Best, V. Claus, E.-R. Olderog; Grundbegriffe der Theoretischen Informatik. Vorlesungsskript, Fachbereich Informatik der Uni Oldenburg. Download möglich. • U. Hedstück, Einführung in die Theoretische Informatik, Oldenbourg, München 2000 • U. Schöning, Theoretische Informatik – kurz gefasst, Akademischer Verlag Heidelberg 1997 • G. Vossen, K.-U. Witt, Grundlagen der Theoretischen Informatik mit Anwendungen, Vieweg, Braunschweig 2000
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Doeben-Henisch, Filip, Rauch, Selder.
Hinweise	

8.4 Unit: Übung Automatentheorie und formale Sprachen

Code	
Modulname	Theoretische Grundlagen der Informatik
Lernform	Übung 1 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	43 h
Anteil Präsenzzeit	17 h
Anteil Selbststudium	26 h
Inhalte	<p>Aufgaben und Beispiele zu den Themen der Unit Vorlesung Automatentheorie und formale Sprachen</p> <p>Die Übungen dienen dazu, dass die Studierenden lernen, eine die algebraischen Fragestellungen der Aufgabe zu verstehen und mit den behandelten Methoden zu lösen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.</p>
Literatur	Siehe Unit Vorlesung Automatentheorie und formale Sprachen
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Doeben-Henisch, Filip, Rauch, Selder.
Hinweise	

9. Modul: Algorithmen und Datenstrukturen

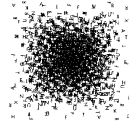
Code	
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP



Prüfungsart	Klausur über 90 Minuten Auf Antrag kann die Prüfung auch in englischer Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Intermediate level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	In dem Modul werden inhaltlich die Module Einführung in die Programmierung mit C, Praxisorientierte Einführung in die Informatik, Grundlagen der Mathematik vorausgesetzt. In den parallel stattfindenden Module Objektorientierte Programmierung mit C++ und Diskrete Mathematik werden die behandelten Themen praktisch umgesetzt bzw. deren mathematische Grundlagen behandelt. Im parallel stattfindenden Modul Theoretische Grundlagen der Informatik werden Teile der hier behandelten Aspekte vom Standpunkt der Automaten und formalen Sprachen behandelt.
Lernergebnis / Kompetenzen	Die mit den Begriffen Algorithmus und abstrakte Datenstruktur verbundenen Kenntnisse sind zentral für die gesamte Informatik Somit werden mit dem Kurs sowohl Schlüsselqualifikationene für die berufliche Tätigkeit als Informatiker, als auch Voraussetzungen für das Verständnis nahezu aller Folgekurse im Verlauf des Studiums vermittelt. Die Studierenden sollen die in de Modul vermittelten Begriffe Algorithmen, Datenstrukturen, Komplexität etc. soweit verstanden haben, dass für einfache bis mittelschwere Problemstellungen <ul style="list-style-type: none"> • geeignete neue Datenstrukturen (aufbauend auf den in dem Kurs behandelten Standardstrukturen) gestaltet werden können • Algorithmen zur Bearbeitung entwickelt und nach den gelernten Methoden dargestellt werden können • Lösungsmöglichkeiten hinsichtlich Korrektheit, Komplexität und Eleganz beurteilt werden können. In den Folgemodulen Informatik sollen die hier vermittelten Begriffe und Techniken selbstverständlich und souverän eingesetzt werden können.
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Algorithmen und Datenstrukturen, Übung Algorithmen und Datenstrukturen
Häufigkeit des Angebots	Nur im Sommersemester
Anerkannte Module	Keine
Modulkoordination	N. N.
Hinweise	

9.1 Unit: Vorlesung Algorithmen und Datenstrukturen

Code	
Modulname	Algorithmen und Datenstrukturen
Lernform	Vorlesung 4 SWS
Sprache	Deutsch
Arbeitsaufwand (h) /	65 h



workload	
Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	<p>Algorithmen:</p> <ul style="list-style-type: none"> • Definition eines Algorithmus und Anforderungen an ihn, semantische Korrektheit • Beschreibungsformen von Algorithmen (natürliche Sprache, Pseudocode, Struktogramm, Datenflußdiagramm, Programmablaufplan) • Komplexitätsbetrachtungen: Platz- und Zeitkomplexität, asymptotische Notationen, Komplexitätsmaße (worst case, average case), P-NP-Problematik • Typen algorithmischer Vorgehensweisen (Rekursion, Greedy, Divide-and Conquer, Backtracking ...) <p>Datenstrukturen:</p> <ul style="list-style-type: none"> • elementare Datenstrukturen • lineare Standardstrukturen (Felder, Listen, Stapel, Warteschlangen) • Bäume • Mengen • Graphen <p>Algorithmen zu den Grundproblemen der Informatik</p> <ul style="list-style-type: none"> • Sortieren • Hashing • Suchen
Literatur	<ul style="list-style-type: none"> • Aho, A. V., Hopcroft, J. E., Ullman, J. D.: The Design and Analysis of Computer Algorithms, Addison-Wesley 1974 • Brunskill, D., Turner, J.: Understanding Algorithms and Data Structures, McGraw-Hill 1996 • Güting, R. H.: Datenstrukturen und Algorithmen, Teubner 1992 • Lewis, T. G., Smith, M. Z.: Datenstrukturen und ihre Anwendung, Oldenbourg 1978 • Mehlhorn, K.: Datenstrukturen und effiziente Algorithmen, Bd 1, 2, 3, Teubner 1986 • Preparata, F. P., Shamos, M. I.: Computational Geometry, Springer 1985 • T. Ottmann, P. Widmayer: Algorithmen und Datenstrukturen. BI Wissenschaftsverlag, Mannheim • H. Reß, G. Vorbeck: Datenstrukturen und Algorithmen. Hanser, München <p>Sedgewick, R.: Algorithms, Addison-Wesley 1984</p>
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine



Lehrende	Hannemann, Kratz, Schubert, Selder, ...
Hinweise	

9.2 Unit: Übung Algorithmen und Datenstrukturen

Code	
Modulname	Algorithmen und Datenstrukturen
Lernform	Übung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	Aufgaben und Beispiele zu den Themen des Moduls Die Übungen dienen dazu, dass die Studierenden lernen, eine Aufgabe zu verstehen und mit den behandelten Methoden zu lösen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Algorithmen und Datenstrukturen
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Hannemann, Kratz, Schubert, Selder, ...
Hinweise	

10. Modul: Vorlesung Objektorientierte Programmierung Grundlagen und Konzepte – C++

Code	
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur am Rechner über 120 Minuten Auf Antrag kann die Prüfung auch in englischer Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Intermediate level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Dieser Modul baut auf das Modul Einführung in Programmierung in C auf. Die zentrale Bedeutung dieses Moduls im Grundstudium Informatik ergibt sich aus der Tatsache, dass hier die grundlegenden Techniken der objektorientierten Programmierung vermittelt werden, die für alle weiteren softwareorientierten Veranstaltungen des Studiums vorausgesetzt werden.
Lernergebnis / Kompetenzen	Die Studierenden erhalten in diesem Modul die grundlegenden Kenntnisse im Bereich der objektorientierten Programmierung. Im Modul werden der Entwurf einer Klassenarchitektur und die verschiedenen Techniken der ob-



	<p>jektorientierten Programmierung vermittelt, die für den aktuellen Stand der Softwaretechnik von zentraler Bedeutung sind. Nach erfolgreichem Abschluss des Moduls sollen die Studierenden</p> <ul style="list-style-type: none"> • die Denk- und Herangehensweise der objektorientierten Programmierung verstanden haben, • zentrale Begriffe wie Datenkapselung und Vererbung erläutern können, • für vorgegebene Aufgabenstellungen <ul style="list-style-type: none"> o einen Klassenentwurf in UML erstellen o die Klassendefinition, die Methoden implementieren und ein Anwendungs-/ Testprogramm implementieren können. • weiterführende Techniken wie Vererbung, Polymorphismus, objektorientierte Ein-/ Ausgabe anwenden können.
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Objektorientierte Programmierung Grundlagen und Konzepte, Übung Objektorientierte Programmierung mit C++
Häufigkeit des Angebots	Nur im Sommersemester
Anerkannte Module	Keine
Modulkoordination	N. N.
Hinweise	

10.1 Unit: Vorlesung Objektorientierte Programmierung Grundlagen und Konzepte –C++

Code	
Modulname	Objektorientierte Programmierung Grundlagen und Konzepte – C++
Lernform	Vorlesung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	65 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	35 h
Inhalte	<ul style="list-style-type: none"> • Objektorientierter Ansatz • Klassenkonzept anhand eines ersten Beispiels, Benennungen, Vergleich von prozeduraler und objektorientierter Programmierung, UML-Darstellung • Initialisierung von Objekten: Konstruktoren, Destruktor, Überladung des Standardkonstruktors, • Arbeiten mit Klassen und Objekten, Anwendung grundlegender Techniken: <ul style="list-style-type: none"> o Trennung von Klassenprogrammierung und Anwendungsprogrammierung o Überladung von Operatoren, insbesondere des Zuweisungsoperators



	<ul style="list-style-type: none"> o Erstellung grundlegender und weiterer Methoden o Dynamische Speicherverwaltung insbesondere bei Zeichenketten • Statische und dynamische (verkettete) Listen • Wiederverwendung von Software, Vererbung • Späte Bindung (Polymorphismus) • Ein- und Ausgabe mit den in C++ vordefinierten Klassen
Literatur	<ul style="list-style-type: none"> • Deitel H.M., Deitel P.J.: C++ How to Program, Prentice Hall • Erlenkötter; Reher: Objektorientiertes Programmieren in C++, rororo • Breymann, Ulrich, C++, Hanser 2001, 6., aktualisierte Aufl
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Alle Informatik Dozenten
Hinweise	

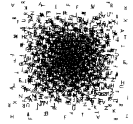
10.2 Unit: Übung Objektorientierte Programmierung mit C++	
Code	
Modulname	Objektorientierte Programmierung Grundlagen und Konzepte - C++
Lernform	Übung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	<p>Die theoretischen Inhalte des Moduls werden an Hand von Aufgaben vertieft und am Rechner umgesetzt</p> <p>Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.</p>
Literatur	Siehe Unit Vorlesung Objektorientierte Programmierung Grundlagen und Konzepte – C++
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	undifferenziert
Lehrende	Alle Informatik Dozenten
Hinweise	



Module des zweiten Studienjahrs

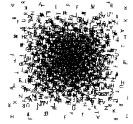
Es ist geplant, jeweils einen der Züge des dritten und vierten Studiensemesters komplett in englischer Sprache durchzuführen. Auf Antrag können in englischsprachlichen Modulen des dritten und vierten Semesters die Prüfungen auch in deutscher Sprache durchgeführt werden.

11. Modul: Software Engineering - Analyse	
Code	Je nach Sprache
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur 90 Minuten Auf Antrag kann die Prüfung in englischer oder deutscher Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Intermediate level course
Notwendige Voraussetzungen	Erfolgreicher Abschluss der Unit Software Engineering – Analyse
Empfohlene Voraussetzungen	Das Modul baut auf den Modulen Einführung in die Programmierung mit C und Objektorientierte Programmierung Grundlagen auf.
Lernergebnis / Kompetenzen	<p>Im Zentrum des Moduls Software Engineering – Analyse stehen die verschiedenen Modelle des Software -Entwicklungsprozesses und die klassische und objektorientierte Analyse der Anforderungen an ein Software -System. Die Studierenden werden mit den Aufgaben von Software -Entwicklern und Projektleitern gleichermaßen vertraut gemacht. Die Studierenden sollen Fähigkeiten zur ingenieurmäßigen Entwicklung von großen Softwaresystemen erlangen. Sie sollen folgende Fragestellungen bearbeiten können:</p> <ul style="list-style-type: none"> • Warum sind qualitativ hochwertige Computer-basierte Systeme notwendig? • Welche Prozeßmodelle können für die Software Entwicklung angewendet werden? • Wie unterscheiden sich die Prozeßmodelle und welche Vor- und Nachteile haben sie? • Welche Anforderungen werden an das Management von Menschen und Entwicklungsprozeß während eines Software Projektes gestellt? • Was sind Software Metriken und wie werden sie genutzt? • Wie können formal Aufwand, Dauer und Risiken eines Software Projektes abgeschätzt werden? • Wie werden Änderungen im Softwareentwicklungsprozess bewältigt? • Wie ist Software im Kontext großer Systeme definiert? • Welche grundsätzlichen Konzepte und Prinzipien werden zur Analyse der Softwareanforderungen angewendet? • Wie werden die klassischen funktions- und datenorientierten Methoden



	<p>und Werkzeuge zur Analyse angewendet?</p> <ul style="list-style-type: none"> • Wie unterscheidet sich die objekt-orientierte Analyse von der klassischen Analyse? • Wie sollten objekt-orientierte Software Projekte geplant und durchgeführt werden? • Welche Methoden und Werkzeuge gibt es in der objekt-orientierten Analyse? <p>Die Studierenden sollen Einsatzmöglichkeiten der verschiedenen Methoden im Anwendungskontext kritisch beurteilen und abschätzen können. Die Studierenden sollen die gestellten Aufgaben in der zugehörigen Unit in kleinen Teams gemeinsam bearbeiten und lösen. Durch die Stärkung der Teamfähigkeit trägt die Unit zum Erwerb der außerfachlichen Kompetenzen bei.</p>
Arbeitsaufwand (h) / Gesamtworkload	150 h 15 h anteiliger Workload zum Erwerb außerfachlicher Kompetenzen
Units	Vorlesung Software Engineering Analyse, Übung Software Engineering Analyse
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	Keine
Modulkoordination	N. N.
Hinweise	

11.1 Unit: Vorlesung Software Engineering - Analyse	
Code	
Modulname	Software Engineering - Analyse
Lernform	Vorlesung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	65 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	<ul style="list-style-type: none"> • Das Software Produkt <ul style="list-style-type: none"> o Die wachsende Rolle der Software o Charakteristiken von Software Produkten • Der Software Entwicklungsprozeß <ul style="list-style-type: none"> o Entwicklungsprozeßmodelle o Software Projektmanagement Konzepte o Prozeß- und Projektmetriken o Software Risikoanalyse und Management o Software Projektplanung o Software Configuration Management • Konventionelles System Engineering



	<ul style="list-style-type: none"> o Geschäftsprozeß Modellierung o Produkt Engineering • Konventionelle Analyse Konzepte <ul style="list-style-type: none"> o Datenorientierte Sicht o Funktionale Sicht • Objektorientierte Analyse Konzepte <ul style="list-style-type: none"> o Unterschiede zum klassischen Paradigma o Methodische Vorgehensweise o Identifikation von Objekten o Management objektorientierter Projekte
Literatur	<ul style="list-style-type: none"> • Balzert, H.: Methoden der objektorientierten Systemanalyse. BI Wissenschaftsverlag, Mannheim, 1995 • Balzert, Helmut: Lehrbuch der Software-Technik. Spektrum Akademischer Verlag, Heidelberg, Bd. 1:2000, Bd. 2:1997 • Peters, James F.; Pedrycz, Witold. Software Engineering - An Engineering Approach. John Wiley & Sons, Inc., New York, NY, 2000 • Pressman, Roger S.: Software Engineering - A Practitioner's Approach 6th ed. McGraw-Hill, New York, NY, 2005 • Rumbaugh, J.; Jacobson, I.; Booch, G.: The Unified Modeling Language Reference Manual. Addison-Wesley, Reading, MA, 1999 • Schach, Stephen R.: Object-Oriented and Classical Software Engineering 5th ed. McGraw-Hill, New York, NY, 2002
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Buhr, Kratz, Wagner, Zöller-Greer
Hinweise	

11.2 Unit: Übung Software Engineering - Analyse	
Code	Je nach Sprache
Modulname	Software Engineering - Analyse
Lernform	Übung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	<p>Die in dem Modul vermittelten Inhalte werden an Hand konkreter Aufgabenstellungen vertieft und praktisch umgesetzt.</p> <p>Die Übungen dienen dazu, dass die Studierenden lernen, die Konzepte der Analyse von Systemen zu verstehen und bei der Lösung konkreter Aufgaben anzuwenden. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unter-</p>



	stützen soll.
Literatur	Siehe Unit Vorlesung Software Engineering Analyse
Art und Form des Leistungsnachweises	Semesterbegleitende Leistungskontrollen durch Übungen am Rechner; Erfolgreiche Teilnahme an den Übungen, das heißt Bestehen aller zur Lehrveranstaltung angebotenen Testate, ist Zulassungsvoraussetzung für die Modulprüfungsleistung.
Bewertung	Undifferenziert
Lehrende	Buhr, Kratz, Wagner, Zöller-Greer
Hinweise	

12. Modul: Statistik

Code	Je nach Sprache
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur über 90 Minuten Auf Antrag kann die Prüfung in englischer Sprache oder deutscher Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Intermediate level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Vorausgesetzt werden die Differential- und Integralrechnung der Schulmathematik, die Lineare Algebra und Abzähltechniken im Umfang der entsprechenden Lehrveranstaltungen im ersten und zweiten Studiensemester.
Lernergebnis / Kompetenzen	Erweiterung des Spektrums an mathematischen Methoden um die sehr praxisrelevanten und aktuellen Verfahren der Statistik, die parallel zur Informatik in vielen Wirtschaftsbereichen Anwendung finden. Es soll ein Grundverständnis für die Begriffe der Wahrscheinlichkeitsrechnung und für den Einsatz und Funktionsweise statistischer Methoden geschaffen werden und deren Einsatz in der Praxis geübt werden. Am Ende der Lehrveranstaltung sollen die Studierenden die wichtigsten Verfahren der schließenden Statistik durchführen können. Anhand der im Rahmen der Lehrveranstaltung behandelten praxisnahen Beispiele sollen die Studierenden befähigt werden, die Ergebnisse statistischer Verfahren interpretieren und beurteilen zu können.
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Statistik, Übung Statistik
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	Keine
Modulkoordination	Behl
Hinweise	



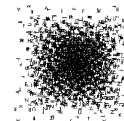
12.1 Unit: Vorlesung Statistik	
Code	
Modulname	Statistik
Lernform	Vorlesung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	65 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	<ul style="list-style-type: none"> • Typische Fragestellungen aus der angewandten Statistik • Grundbegriffe der Statistik • Verfahren der beschreibenden Statistik (Kenngrößen, graphische Darstellungsmethoden) • Wahrscheinlichkeitsrechnung (Zufallsvariable, Unabhängigkeit, bedingte Wahrscheinlichkeit, Formel von Bayes, Verteilungen) • Umgang mit ausgewählten Verteilungen (Normalverteilung, Exponentialverteilung, Weibullverteilung und deren Zufallsstrebereiche) • Grundlagen der schließenden Statistik (Grundgesamtheit, Stichprobe, Hypothesen, Schließen bei unvollständiger Information, α-, β-Fehler, statistische Signifikanz) • Schätzen von Parametern und Konfidenzintervalle • ausgewählte statistische Testverfahren (Mittelwertvergleich, einfache Varianzvergleich, Verteilungstests) • graphische Verfahren der schließenden Statistik (Wahrscheinlichkeitsnetz, Weibull-Netz)
Literatur	<ul style="list-style-type: none"> • Fersch: Beschreibende Statistik, Physica-Verlag. • Mayer: Beschreibende Statistik, Hanser Verlag. • Hartung: Statistik, Lehr und Handbuch der angewandten Statistik, Oldenbourg. • Toutenburg: Deskriptive Statistik, Springer Verlag. • Dürr, Mayer: Wahrscheinlichkeitsrechnung und Schließende Statistik, Hanser Verlag. • Bley Müller, Gehlert, Gülicher: Statistik für Wirtschaftswissenschaftler, Verlag Vahlen. • Basler: Grundbegriffe der Wahrscheinlichkeitsrechnung und Statistischen Methodenlehre, Physica-Verlag. • Greiner, Tinhofer: Stochastik für Studienanfänger der Informatik, Hanser Verlag. • Henze: Stochastik für Einsteiger, Vieweg Verlag. • Lehn, Wegmann: Einführung in die Statistik, Teubner Verlag. • Beichelt: Stochastik für Ingenieure, Teubner Verlag. • Clauß, Finze, Partzsch: Statistik für Soziologen, Pädagogen, Psycholo-



	<p>gen und Mediziner, Grundlagen, Verlag Harry Deutsch.</p> <ul style="list-style-type: none"> • Pitman: Probability, Springer Verlag. • Krämer: So lügt man mit Statistik, Campus Verlag. • Orth, Soravia: Design of Experiments in Ullman's Encyclopedia of Industrial Chemistry, Wiley-VCH. • de Mehr, Orth, Altenschmidt: Statistische Methoden zur Qualitätssicherung für die chemische Industrie. Lehrgang der DGQ. • Rinne, Mittag: Statistische Methoden der Qualitätssicherung, Hanser Verlag.
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Behl, Falkenberg, Orth
Hinweise	

12.2 Unit: Übung Statistik	
Code	
Modulname	Statistik
Lernform	Übung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	<p>Aufgaben und Beispiele aus den im Modul behandelten Themen</p> <p>Die Übungen dienen dazu, dass die Studierenden lernen, die statistischen Fragestellungen der Aufgabe zu verstehen und mit den behandelten Methoden zu lösen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.</p>
Literatur	Siehe Unit Vorlesung Statistik
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Behl, Falkenberg, Orth
Hinweise	

13. Modul: Objektorientierte Programmierung Vertiefung - Java	
Code	Je nach Sprache
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Eine eigene erfolgreich durchgeführte Applikationsentwicklung mit voll-



	ständiger Dokumentation Auf Antrag kann die Prüfung in englischer Sprache oder deutscher Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Advanced level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Das Modul vertieft die im Modul Objektorientierte Programmierung Grundlagen erworbenen Kenntnisse und Fähigkeiten. Das parallel stattfindende Modul Software-Engineering – Analyse vermittelt die hier praktisch umzusetzenden Software Engineering Kenntnisse.
Lernergebnis / Kompetenzen	Dieser Modul führt das Modul Objektorientierte Programmierung Grundlagen fort. Die Studierenden sollen hier die Fähigkeit zum Entwurf und zur Programmierung von komplexeren, ereignisgesteuerten GUI-Applikationen erlernen. Dabei wird als Programmiersprache JAVA eingesetzt. Dieser Modul ist eine wichtige praktische Vorbereitung für die sich daran anschließenden Veranstaltungen zum Thema Software-Engineering und Projekt. Team- und Projektarbeit wird hier ausdrücklich eingeübt und für die Studierenden erfahrbar gemacht. Die in der Lehrveranstaltung Objektorientierte Programmierung mit C++ erworbenen Kenntnisse sollen erweitert und vertieft werden. Am Ende der Lehrveranstaltung sollen die Studierenden in der Lage sein große objektorientierte ereignisgesteuerte GUI-Applikationen zu entwerfen und zu implementieren.
Arbeitsaufwand (h) / Gesamtworkload	150 h 15 h anteiliger Workload zum Erwerb außerfachlicher Kompetenzen
Units	Vorlesung Objektorientierte Programmierung Vertiefung Java, Übung Objektorientierte Programmierung Java
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	Keine
Modulkoordination	Wagner
Hinweise	

13.1 Unit: Vorlesung Objektorientierte Programmierung Vertiefung - Java

Code	
Modulname	Objektorientierte Programmierung Vertiefung - Java
Lernform	Vorlesung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	65 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	<ul style="list-style-type: none"> • Verwendung von Klassenbibliotheken • Graphische Dialogelemente aller Art



	<ul style="list-style-type: none"> • Spezifikationen aus Anwendersicht • DV-Spezifikationen • Entwurf und Realisierung von Datei-gestützten Dialogen • Entwurf und Realisierung von Anwendungen, die aus mehreren Teildialogen mit gemeinsamer Kommunikationsstrukturen bestehen.
Literatur	<ul style="list-style-type: none"> • Gosling, J.; Joy, B.; Steele, G.: The Java Language Specification., Addison-Wesley, Reading, MA, 1996 • Liskov, B.; Guttag, J.: Program Development in Java. Addison-Wesley, Boston, 2001
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Kratz, Schubert, Wagner
Hinweise	

13.2 Unit: Übung Objektorientierte Programmierung Java	
Code	
Modulname	Objektorientierte Programmierung Vertiefung - Java
Lernform	Übung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	In den Übungen werden im Laufe des Semesters von den Studierenden zunächst einfache, später komplexere objektorientierte ereignisgesteuerte GUI-Applikationen entwickelt und implementiert. Die Übungen dienen dazu, die im ersten Studienjahr und im Laufe der Vorlesung Objektorientierte Programmierung Vertiefung erworbenen Kenntnisse und Fähigkeiten der Objektorientierten Programmierung praktisch anzuwenden und zu vertiefen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Objektorientierte Programmierung Java
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Kratz, Schubert, Wagner
Hinweise	

14. Modul: Datenbanken	
Code	Je nach Sprache
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester



Credits	5 CP
Prüfungsart	Klausur über 120 Minuten Auf Antrag kann die Prüfung in englischer Sprache oder deutscher Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Intermediate level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Inhaltlich baut das Modul auf Kenntnissen der Algebra aus dem Modul Grundlagen der Mathematik, der Programmierung aus den entsprechenden Modulen sowie des Software-Engineering aus dem gleichnamigen Modul auf.
Lernergebnis / Kompetenzen	Datenbanken spielen in allen Anwendungen der Informatik eine zentrale Rolle. Mit dem Modul werden Kernkompetenzen der Informatik vermittelt. Die Veranstaltung vermittelt Kenntnisse und Fähigkeiten, die die Studierenden in die Lage versetzen, qualifizierte Beiträge zur Gestaltung und Nutzung von Datenbanken als zentraler Basis betrieblicher Informationsverarbeitung zu leisten. Im Mittelpunkt stehen dabei relationale Datenbanken: die Studierenden sollen fundiertes Wissen über das relationale Datenmodell und seine SQL-Implementierung erwerben. und dieses in praktischen Übungen an einem konkreten Datenbankmanagementsystem umsetzen.
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Datenbanken, Übung Datenbanken
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	Keine
Modulkoordination	Schubert
Hinweise	

14.1 Unit: Vorlesung Datenbanken	
Code	
Modulname	Datenbanken
Lernform	Vorlesung 4 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	90 h
Anteil Präsenzzeit	70 h
Anteil Selbststudium	20 h
Inhalte	<ul style="list-style-type: none"> • Konzeptionelle Grundlagen : Datenbankkonzept, Datenbankarchitektur, Datenmodelle • Das relationale Modell: Datenmodell, Strukturelle Integritätsbedingungen, Relationen-Algebra, Datenbankschema • Die relationale Datendefinitions- und -manipulationssprache SQL



	<ul style="list-style-type: none"> • Datenmodellierung und Relationales Datenbankschema: Entity-Relationship-Modell, Normalisierungsverfahren, Historische Datenhaltung • Systemarchitektur: Systemkataloge, Benutzerverwaltung, Transaktionsverwaltung • Elemente der Datenbankprogrammierung: Formulare, Ereignis-Steuerung, Datenbank-Prozeduren , Datenbank-Schnittstellen
Literatur	<ul style="list-style-type: none"> • Abbey/Corey/Abramson: Oracle 9i. A Beginner's Guide, Osborne/McGraw-Hill • Date: An Introduction to Database Systems, Addison-Wesley • Connolly/Begg: Database Systems: A Practical Approach to Design, Implementation and Management, Addison-Wesley • Elmasri/Navathe: Fundamentals of Database Systems, Benjamin/Cummings • Härder/Rahm: Datenbanksysteme - Konzepte und Techniken der Implementierung, Springer • Heuer/Saake: Datenbanken, Thomson • Heuer, Objektorientierte Datenbanken, Addison-Wesley • Kemper/Eickler: Datenbanksysteme, Oldenbourg • Kuhlmann/Müllmerstadt: SQL – Eine strukturierte Einführung, rororo • Loney/Koch: Oracle 9i, The Complete Reference, Oracle Press • Meier: Relationale Datenbanken, Springer • Meier/Wüst: Objektorientierte und objektrelationale Datenbanken, dpunkt.verlag • Panny/Taudes: Einführung in den Sprachkern von SQL-99, Springer • Pribyl/Feuerstein: Learning Oracle PL/SQL, O'Reilly • Raymans: Oracle Programmierung – Datenbankprogrammierung und –administration, Addison-Wesley • Schicker: Datenbanken und SQL, Teubner • Schubert, M.: Datenbanken – Theorie, Entwurf und Programmierung relationaler Datenbanken, Teubner Verlag
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Buhr, Hackenbracht, Lindner, Schubert
Hinweise	

14.2 Unit: Übung Datenbanken	
Code	
Modulname	Datenbanken
Lernform	Übung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul



Arbeitsaufwand (h) / workload	60 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	25 h
Inhalte	Im Mittelpunkt der Übungen stehen relationale Datenbanken: die Studierenden sollen das in der Vorlesung vermittelte Wissen über das relationale Datenmodell und seine SQL-Implementierung in praktischen Übungen an einem konkreten Datenbankmanagementsystem umsetzen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Datenbanken
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Buhr, Hackenbracht, Lindner, Schubert
Hinweise	

15. Modul: Rechnernetze

Code	Je nach Sprache
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur über 90 Minuten Auf Antrag kann die Prüfung in englischer Sprache oder deutscher Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Intermediate level course
Notwendige Voraussetzungen	Erfolgreicher Abschluss der Unit Übung Rechnernetze
Empfohlene Voraussetzungen	Inhaltlich wird auf den Modulen Praxisorientierte Einführung in die Informatik, Einführung in die Programmierung mit C und Theoretische Grundlagen der Informatik aufgebaut.
Lernergebnis / Kompetenzen	Der schnelle Zugriff auf Informationen und Wissen ist inzwischen ein entscheidender Wettbewerbsfaktor. Immer weniger Arbeitsplätze kommen ohne Computerunterstützung aus. Durch Rechnernetze sind Fachabteilungen und Unternehmensstandorte untereinander und weltweit mit Kunden und Lieferanten verbunden. Das Wissen über die Funktionsweise von Computersystemen und deren Verbund über Rechnernetze hilft dem Informatiker zum Beispiel ständig neue Anwendungsfelder zu erschließen. Somit werden mit Hilfe dieses Moduls die wichtigen analytischen, kreativen und gestalterischen Fähigkeiten erweitert. Die Studierenden sollen grundlegende Konzepte und unterschiedlichen Funktionsweisen von Rechnernetzen und deren Nutzung kennenlernen und verstehen.
Arbeitsaufwand (h) /	150 h



Gesamtworkload	
Units	Vorlesung Rechnernetze, Übung Rechnernetze
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	Keine
Modulkoordination	Filip
Hinweise	

15.1 Unit: Vorlesung Rechnernetze	
Code	
Modulname	Rechnernetze
Lernform	Vorlesung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	65 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	<p>Im Rahmen der Lehrveranstaltung werden grundlegende Inhalte und Methoden zu folgenden Themen behandelt. Dies sind im Einzelnen:</p> <ul style="list-style-type: none"> • Einführung in Rechnernetze • Datenübermittlung • OSI - Referenzmodell • Lokale Netze • LAN - Erweiterungen • Internetworking • Netzwerkmanagement • IPv6
Literatur	<ul style="list-style-type: none"> • Andrew S. Tannenbaum, Computer Netzwerke, Third Edition, Pearson, 2000. • Douglas E. Comer, Computernetzwerke und Internets, Pearson, 2001. • Douglas E. Comer, Internetworking with TCP/IP, Volume I, Prentice Hall, 1995 • James F. Kurose, Keith W. Ross, Computernetze, Pearson, 2002 • Karanjit S. Siyan, Inside TCP/IP, Third Edition, New Riders Publishing, 1997. • Anatol Badach, Erwin Hoffmann, Technik der IP-Netze, Hanser, 2000 • Wolfgang Böhmer, VPN – Virtual Private Networks, Hanser, 2002 • Herbert Wiese, Das neue Internetprotokoll IPv6, Hanser 2002
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Filip, Kappes



Hinweise	
----------	--

15.2 Unit: Übung Rechnernetze	
Code	
Modulname	Rechnernetze
Lernform	Übung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	Die Übungen dienen dazu, dass die Studierenden lernen, die theoretischen Inhalte zu verstehen und anzuwenden. Ein wesentliches Element der Übungen und des Labors ist die praktische Lösung der gestellten Aufgaben im Rechnernetze-Labor. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Rechnernetze
Art und Form des Leistungsnachweises	Teilnahme an mindestens 80% aller Übungen ist Zulassungsvoraussetzung für die Modulprüfungsleistung
Bewertung	Keine
Lehrende	Filip, Kappes
Hinweise	

16. Modul: Betriebssysteme	
Code	Je nach Sprache
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur über 90 Minuten Auf Antrag kann die Prüfung in englischer Sprache oder deutscher Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Intermediate level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Grundlagen sind die Informatikmodule der ersten beiden Studiensemester.
Lernergebnis / Kompetenzen	Der schnelle und sichere Zugriff auf Informationen und Wissen ist inzwischen ein entscheidender Wettbewerbsfaktor. Immer weniger Arbeitsplätze kommen ohne Computerunterstützung aus. Das Wissen über die Funktionsweise von Computersystemen, speziell die Funktionsweise von Betriebssystemen hilft dem Informatiker zum Beispiel ständig neue Anwendungsfelder zu erschließen. Somit werden mit Hilfe dieses Moduls die



	wichtigen analytischen, kreativen und gestalterischen Fähigkeiten erweitert. Die Studierenden sollen grundlegende Konzepte und Verfahren zur Realisierung von Betriebssystemen kennenlernen und verstehen.
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Betriebssysteme, Übung Betriebssysteme
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	Keine
Modulkoordination	Filip
Hinweise	

16.1 Unit: Vorlesung Betriebssysteme	
Code	
Modulname	Betriebssysteme
Lernform	Vorlesung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	65 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	Im Rahmen der Lehrveranstaltung werden grundlegende Inhalte und Methoden zu folgenden Themen behandelt. Dies sind im Einzelnen: <ul style="list-style-type: none"> • Prozesse und Prozeßverwaltung • Interprozeßkommunikation • Speicherverwaltung • Dateisystem • Ein- und Ausgabegeräte • Verteilte Betriebssysteme • Windows und Unix als konkrete Betriebssysteme
Literatur	<ul style="list-style-type: none"> • A.S. Tanenbaum, Moderne Betriebssysteme, Pearson 2002. • A. Silberschatz, P.B. Galvin, Operating System Concepts, 4th ed., Addison-Wesley, 1998. • Stallings, Operating Systems, 4th ed, Prentice-Hall, Inc., 2001 • R. Brause, Betriebssysteme – Grundlagen und Konzepte, Springer, 1998. • Vogt, C., Betriebssysteme, Spektrum, Akad. Verl. 2001. • Nehmer, J., Sturm, P., Systemsoftware, dpunkt Verlag, 1998.
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Filip, Kappes, Klingemann, Selder



Hinweise	
----------	--

16.2 Unit: Übung Betriebssysteme	
Code	
Modulname	Betriebssysteme
Lernform	Übung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	Die Übungen dienen dazu, dass Theorie und Praxis von Betriebssystemen zu verstehen und konkrete Aufgabenstellungen praktisch zu lösen und umzusetzen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Betriebssysteme
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Filip, Kappes, Klingemann, Selder
Hinweise	

17. Modul: Numerische Mathematik	
Code	Je nach Sprache
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur über 90 Minuten. Auf Antrag kann die Prüfung in englischer Sprache oder deutscher Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Basic level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Das Modul setzt Kenntnisse der linearen Algebra (Vektor- und Matrizenrechnung, lineare Gleichungssysteme) voraus, wie sie in dem Modul Algebra im 1. Semester vermittelt werden. Beherrschung der Analysis, wie sie im Rahmen der Schulausbildung vermittelt wird, ist in hohem Maße zweckdienlich. Außerdem sind umfangreichere Programmiererfahrungen in einer höheren Programmiersprache wünschenswert.
Lernergebnis / Kompetenzen	In dem Modul werden grundlegende numerische Aufgabenstellungen vermittelt sowie verschiedene Verfahren zu deren Lösung entwickelt. Diese Verfahren werden für den Einsatz auf Computern algorithmisch aufbereitet



	<p>und hinsichtlich verschiedener Kriterien wie Rechenaufwand und Genauigkeit analysiert.</p> <p>Die Studierenden sollen in der Lage sein, für konkrete Problemstellungen geeignete Verfahren auszuwählen und darauf aufbauend Programme zu deren Bearbeitung zu entwickeln.</p> <p>Das Modul trägt damit zum Ausbau der Methodenkompetenz eines Informatikers bei. Am Ende der Lehrveranstaltung sollen die Studierenden die Lösungsverfahren für Grundaufgaben der numerischen Mathematik sicher beherrschen und anwenden können</p>
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Numerische Mathematik, Übung Numerische Mathematik
Häufigkeit des Angebots	Nur im Sommersemester
Anerkannte Module	Keine
Modulkoordination	Wolf
Hinweise	

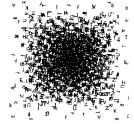
17.1 Unit: Vorlesung Numerische Mathematik	
Code	
Modulname	Numerische Mathematik
Lernform	Vorlesung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	65 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	<p>In dem Modul werden computergerechte numerische Verfahren vermittelt. Im Vordergrund stehen die Grundideen zur algorithmischen Lösung wichtiger numerischer Aufgaben. Dabei geht es auch um die Frage, inwieweit man den computergenerierten Ergebnissen trauen kann. In praktischen Anwendungen sollen Genauigkeit, Effizienz und Verlässlichkeit numerischer Verfahren untersucht werden.</p> <p>Das Modul widmet sich im einzelnen den folgenden Themen:</p> <ul style="list-style-type: none"> • Algorithmusbegriff, Zahldarstellung, numerische Fehler und Fehlerfortpflanzung • Nichtlineare Gleichungen • Lineare Gleichungssysteme • Nichtlineare Gleichungssysteme • Numerische Integration • Differenzenverfahren • Interpolation und Approximation • Freiformkurven • Optimierungsverfahren



	<ul style="list-style-type: none"> • Überblick über numerische Software (Computeralgebrasysteme und Numerik-Bibliotheken)
Literatur	<ul style="list-style-type: none"> • Schwarz, H.R.: Numerische Mathematik. Teubner Verlag • Stoer, J., Bulirsch, R.: Einführung in die numerische Mathematik. Springer-Verlag • Opfer, G.: Numerische Mathematik für Anfänger. Vieweg Verlag
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Birkholz, Wolf
Hinweise	

17.2 Unit: Übung Numerische Mathematik	
Code	
Modulname	Numerische Mathematik
Lernform	Übung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	Die in dem Modul vermittelten Inhalte werden an Hand konkreter Aufgabenstellungen vertieft und praktisch umgesetzt. Die Übungen sind in Form eines Numerik-Praktikums am PC zu bearbeiten.
Literatur	Siehe Unit Vorlesung Numerische Mathematik
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Undifferenziert
Lehrende	Birkholz, Wolf
Hinweise	

18. Modul: Software Engineering – Design	
Code	Je nach Sprache
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur 90 Minuten; Auf Antrag kann die Prüfung in englischer Sprache oder deutscher Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Advanced level course
Notwendige Voraussetzungen	Keine



Empfohlene Voraussetzungen	Das Modul baut auf den Modulen der Programmierung sowie dem Modul Software Engineering - Analyse auf.
Lernergebnis / Kompetenzen	<p>Im Zentrum des Moduls stehen die Software - Entwicklungsphasen Entwurf, Implementierung und Wartung. Klassisches und objekt-orientierter Entwurf werden gegenübergestellt. Im Rahmen des objekt-orientierten Designs wird der Entwurf mit Hilfe Pattern-orientierter Software Architektur eingeführt.</p> <p>Die Studierenden sollen Fähigkeiten zur ingenieurmäßigen Entwicklung von großen Softwaresystemen erlangen. Sie sollen folgende Fragestellungen bearbeiten können:</p> <ul style="list-style-type: none"> • Welche grundsätzlichen Prinzipien und Konzepte werden zum Entwurf von Software eingesetzt? • Warum ist eine Software Architektur notwendig? • Welche Entwurfsmodelle für Daten, Architektur, Schnittstellen und Komponenten gibt es ? • Was sind die grundlegenden Konzepte und Prinzipien des objektorientierten Designs? • Wie funktioniert der Übergang von der objektorientierten Analyse zum objektorientierten Entwurf? • Wie wurden die Methoden des objektorientierten Entwurfs vereinheitlicht? • Wie wird die Unified Modeling Language angewendet? • Was sind die Elemente des Software Designs mit Mustern und wie werden sie eingesetzt? • Wie wird komponenten-basiertes Software Engineering verwendet, um Systeme aus wiederverwendbaren Komponenten zu entwickeln? • Wie werden Client-Server - Systeme entworfen? • Welche technischen Aktivitäten werden während eines Cleanroom Software Engineering Prozesses durchgeführt? • Wie wird Software formal spezifiziert? • Welche fortgeschrittenen Software Engineering Konzepte gibt es und was sind ihre Einsatzschwerpunkte? <p>Die Studierenden sollen Einsatzmöglichkeiten der verschiedenen Methoden im Anwendungskontext kritisch beurteilen und abschätzen können.</p> <p>Die Studierenden sollen die gestellten Aufgaben in kleinen Teams gemeinsam bearbeiten und lösen. Durch die Stärkung der Teamfähigkeit trägt die Unit zum Erwerb der außerfachlichen Kompetenzen bei.</p>
Arbeitsaufwand (h) / Gesamtworkload	150 h 15 h anteiliger Workload zum Erwerb außerfachlicher Kompetenzen
Units	Vorlesung Software Engineering – Design, Übung Software Engineering – Design
Häufigkeit des Angebots	Nur im Sommersemester
Anerkannte Module	Keine



Modulkoordination	Wagner
Hinweise	

18.1 Unit: Vorlesung Software Engineering – Design	
Code	
Modulname	Software Engineering – Design
Lernform	Vorlesung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	65 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	<ul style="list-style-type: none"> • Der Software Design Prozess <ul style="list-style-type: none"> o Software Design und Software Qualität • Software Design Prinzipien • Software Design Konzepte <ul style="list-style-type: none"> o Abstraktion o Modularität o Software Architektur o Kontrollhierarchie o Datenstruktur • Software Architektur <ul style="list-style-type: none"> o Datenstrukturen o Alternative Architekturentwürfe o Abbildung der Anforderungen in die Architektur • Objektorientiertes Software Design <ul style="list-style-type: none"> o Unterschiede zum klassischen Paradigma o Vereinheitlichte Vorgehensweise - UML • System Design Prozess • Objekt Design Prozess • Software Design mit Mustern (Patterns) • Software Test <ul style="list-style-type: none"> o Testbarkeit als Entwurfsziel o Software Test Strategie o Software Test Techniken o Objektorientiertes Testen • Weiterführende Methoden des Software Engineering <ul style="list-style-type: none"> o Formale Methoden o Cleanroom Software Engineering o Komponenten-basiertes Software Engineering



	<ul style="list-style-type: none"> o Client-Server Software Engineering o Software Reengineering <p>Das Modul Realzeitsysteme zeigt anschauliche Anwendungen bzw. Erweiterungen des Moduls.</p>
Literatur	<ul style="list-style-type: none"> • Balzert, H.: Methoden der objektorientierten Systemanalyse. BI Wissenschaftsverlag, Mannheim, 1995 • Balzert, Helmut: Lehrbuch der Software-Technik. Spektrum Akademischer Verlag, Heidelberg, Bd. 1:2000, Bd. 2:1997 • Buschmann, F., et. al.. Pattern Oriented Software Architecture -A System of Patterns. John Wiley & Sons, Inc., New York, NY, 1996 • Gamma E., et. al. Design Patterns - Elements of Reusable Object Oriented Software. Addison-Wesley, Reading, MA, 1995 • Peters, James F.; Pedrycz, Witold. Software Engineering - An Engineering Approach. John Wiley & Sons, Inc., New York, NY, 2000 • Pressman, Roger S.: Software Engineering - A Practitioner's Approach 6th ed. McGraw-Hill, New York, NY, 2005 • Rumbaugh, J.; Jacobson, I.; Booch, G.: The Unified Modeling Language Reference Manual. Addison-Wesley, Reading, MA, 1999 • Schach, Stephen R.: Object-Oriented and Classical Software Engineering 5th ed. McGraw-Hill, New York, NY, 2002
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Kratz, Wagner, Zöller-Greer
Hinweise	

18.2 Unit: Übung Software Engineering – Design	
Code	
Modulname	Software Engineering – Design
Lernform	Übung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	Die Übungen dienen dazu, dass die in der Vorlesung behandelten Konzepte und Methoden des Designs von Software zu verstehen und umzusetzen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Software Engineering -- Design
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Kratz, Wagner, Zöller-Greer



Hinweise	
----------	--

19. Modul: Echtzeitsysteme	
Code	Je nach Sprache
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur über 90 Minuten Auf Antrag kann die Prüfung in englischer Sprache oder deutscher Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Advanced level course
Notwendige Voraussetzungen	Erfolgreicher Abschluss der Unit Übung Echtzeitlabor
Empfohlene Voraussetzungen	Inhaltlich baut das Modul auf die Module Einführung in die Programmierung mit C, Grundlagen der Informatik und maschinennahe Programmierung, Theoretische Grundlagen der Informatik (Rechnerarchitektur) und Betriebssysteme auf. Unter anderem ergänzt dieses Modul die Module Datenbanken und Rechnernetze im Bereich zeitabhängiger Vorgänge.
Lernergebnis / Kompetenzen	In vielen Bereichen der Programmierung technischer und kommerzieller Systeme spielen zeitabhängige Vorgänge eine wichtige Rolle. Hierzu gehören auch die Inter-Prozess-Kommunikation und die Programmierung von Schnittstellen zur Kommunikation mit externen Geräten. In diesem Modul wird die Fähigkeit vermittelt, exemplarische Systeme zu modellieren und in lauffähige Programme umzusetzen. Ziel ist die Fähigkeit, zeitabhängige Vorgänge sowohl innerhalb eines Rechners wie auch bei der Kommunikation mit externen Geräten zu modellieren und zu programmieren.
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Echtzeitsysteme, Übung Echtzeitlabor
Häufigkeit des Angebots	Nur im Sommersemester
Anerkannte Module	keine
Modulkoordination	Güsmann
Hinweise	

19.1 Unit: Vorlesung Echtzeitsysteme	
Code	
Modulname	Echtzeitsysteme
Lernform	Vorlesung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	65 h



Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	<p>Gegenstand der Vorlesung sind die Entwicklung zeitabhängiger Systeme und Grundprinzipien der Prozesssteuerung mit Digitalrechnern. Einzelne Themen sind nachfolgend aufgelistet:</p> <ul style="list-style-type: none"> • Deterministische und stochastische Prozesse, synchrone und asynchrone Events, Echtzeitverhalten • Modellierung von Prozessen mit Entwurfswerkzeugen • Parallelisierung und Synchronisation • Inter-Prozess-Kommunikation • Zuverlässigkeit, Redundanz, Fehlertoleranz • Betriebssysteme für Echtzeitprogrammierung • Bussysteme für Echtzeitrechner • Prozesshardware (Sensoren und Aktoren), • Schnittstellenprogrammierung • A/D- und D/A-Konverter
Literatur	<ul style="list-style-type: none"> • Stuart Bennett: Real-Time Computer Control, Prentice Hall, 1994 • Erik Jacobson: Einführung in die Prozeßdatenverarbeitung, München, Hanser 1996 • Walter Jakoby: Automatisierungstechnik. Algorithmen und Programme, Berlin, Springer 1996 • Walter Motsch: Prozeßrechnerstrukturen, Braunschweig, Vieweg 1995 • Profos/Pfeifer: Grundlagen der Meßtechnik, München, Oldenbourg 1997 • Eckehard Schnieder: Prozeßinformatik, Braunschweig, Vieweg 1993 • Schiffmann/Schmitz: Technische Informatik Bd. 1 und 2, Berlin, Springer 2001
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Güsmann, Rauch, Döben-Henisch
Hinweise	

19.2 Unit: Übung Echtzeitlabor	
Code	
Modulname	Echtzeitsysteme
Lernform	Übung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	Aus den in dem Modul behandelten Themen werden Aufgaben aus Theorie



	und Praxis gestellt.
Literatur	Siehe Unit Vorlesung Echtzeitsysteme
Art und Form des Leistungsnachweises	Erfolgreiche Teilnahme an den Übungen, das heißt Bestehen aller zur Lehrveranstaltung angebotenen Testate, ist Zulassungsvoraussetzung für die Modulprüfungsleistung. Die Übungen dienen dazu, dass Theorie und Praxis von Echtzeitsystemen zu verstehen und konkrete Aufgabenstellungen praktisch zu lösen und umzusetzen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Bewertung	Undifferenziert
Lehrende	Güsmann, Rauch, Döben-Henisch
Hinweise	

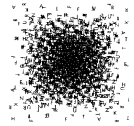
20. Modul: Datenschutz / IT Security	
Code	Je nach Sprache
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur über die Inhalte der Units Datenschutz und IT Security 120 Minuten Auf Antrag kann die Prüfung in englischer Sprache oder deutscher Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Advanced level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Das Modul umfasst die beiden inhaltlich stark verzahnten Themen Datenschutz und IT Security. Inhaltlich wird bei der Behandlung des Themas IT Security auf den Modulen, Grundlagen der Informatik und maschinennahe Programmierung, Einführung in die Programmierung mit C, Theoretische Grundlagen der Informatik, Rechnernetze, Betriebssysteme aufgebaut.
Lernergebnis / Kompetenzen	Im Umfeld einer globalen arbeitsteiligen Wirtschaft sind vernetzte Computersysteme unabdingbare Infrastrukturen. Ihr sicheres und schnelles Funktionieren ist kritisch für den unternehmerischen Erfolg. Auch für die Wissenschaft und die Gesellschaft sind sichere und zuverlässige Kommunikationsdienste unverzichtbar. Durch das Erlernen der hierzu benötigten Konzepte und Architekturen zur sicheren Datenübertragung lassen sich neue Anwendungsfelder erschließen. Somit werden mit Hilfe dieses Moduls die wichtigen analytischen, kreativen und gestalterischen Fähigkeiten erweitert Ein wesentlicher Aspekt in diesem Zusammenhang stellt der Datenschutz dar. Den Studierenden werden in dem Modul spezifische juristische Grundkenntnisse sowie in ausgesuchten Bereichen vertieftes Fachkenntnis, die für den Datenschutz besonders relevant sind, vermittelt. Hinzu



	<p>kommt die Befassung mit aktuellen Rechtsproblemen, die im Zusammenhang mit der Nutzung von Internet Anwendungen entstehen oder die durch Anwendungen der Informationstechnik ausgelöst werden. Das Modul trägt zum Verständnis der Bedeutung der IT für das Unternehmen und die Gesellschaft bei und fördert die Fähigkeit zu verantwortlicher Arbeit in Kooperation mit anderen.</p> <p>Lernziele sind:</p> <ul style="list-style-type: none"> • Vermittlung der für die Verarbeitung personenbezogener Daten relevanten Rechtsnormen • Kenntnis und Verständnis von Datenschutz- und Sicherheitsrisiken vernetzter Computersysteme sowie grundlegender Konzepte, Architekturen zum Aufbau und Betreiben sicherer Netze
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Datenschutz, IT Security
Häufigkeit des Angebots	Nur im Sommersemester
Anerkannte Module	Keine
Modulkoordination	N. N.
Hinweise	

20.1 Unit: Vorlesung Datenschutz

Code	
Modulname	Datenschutz / IT Security
Lernform	Vorlesung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	75 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	40 h
Inhalte	<p>Gegenstand der Lehrveranstaltung Datenschutz sind grundlegende Themenfelder aus den Bereichen Datenschutz und Datensicherheit:</p> <ul style="list-style-type: none"> • Bundesdatenschutzgesetz (BDSG) und ausgesuchte Landesdatenschutzgesetze • Voraussetzungen der Verarbeitung personenbezogener Daten (Zulässigkeit der Datenverarbeitung, -speicherung und -übermittlung) • Rechte der Betroffenen (Benachrichtigung, Auskünfte, Lösungs- und Korrekturrechte) • Erforderliche technische und organisatorische Maßnahmen zur Datensicherung • Kontrolle der Einhaltung des Datenschutzes durch betriebliche und staatliche Kontrollinstanzen • Funktion und Wirkungsweise von Datensicherungskonzepten • Datenschutz im internationalen Bereich (EU-Recht, Safe harbour prin-



	cipals u.ä.)
Literatur	<ul style="list-style-type: none"> • Tinnefeld/Ehmann, Einführung in das deutsche Datenschutzrecht, 3. Auflage München / Wien 1999 • Schaar, Datenschutz im Internet, München 2002 • Standardkommentare zum BDSG (u.a. Simitis; Gola; Schaffland/Wiltfang; Däubler/Klebe/Wedde, jeweils in neuester Auflage) • Grundschutzhandbuch des Bundesamtes für Sicherheit in der Informationstechnik (Online-Dokument) • Wedde/ Schröder, Das Gütesiegel für Qualität im betrieblichen Datenschutz, 2001
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Wedde
Hinweise	

20.2 Unit: Vorlesung IT Security

Code	
Modulname	Datenschutz / IT Security
Lernform	Vorlesung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	75 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	40 h
Inhalte	<p>Im Rahmen der Lehrveranstaltung IT Security werden grundlegende Inhalte und Methoden zu folgenden Themen behandelt:</p> <ul style="list-style-type: none"> • Grundlagen - Netze • Sicherheitsrisiken • Firewalls • Kryptographie • Systeme und Anwendungen
Literatur	<ul style="list-style-type: none"> • O. Kyas, „Sicherheit im Internet“, DATAKOM, 1998. • S. Strobel, „Firewalls für das Netz der Netze“, dpunkt-Verlag, 1997. • M. Raeppe, „Sicherheitskonzepte für das Internet“, dpunkt.verlag, 1998. • K. Schmech, „Kryptografie und Public-Key-Infrastrukturen im Internet“, 2. Auflage, dpunkt.verlag, 2001. • D.B. Chapman/E.D. Zwicky, „Building Internet Firewalls“, O'Reilly & Associates, 2. Auflage, 2000
Art und Form des Leistungsnachweises	Kein Leistungsnachweis



Bewertung	Keine
Lehrende	Filip, Kappes
Hinweise	

21. Modul: IT Projekte	
Code	Je nach Sprache
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	10 CP
Prüfungsart	In dem Modul sind drei Teilprüfungsleistungen in den Units IT Projektmanagement, Fallstudie Prozessmodellierung und Programmierpraktikum mit Datenbanken zu erbringen. Die Note des Moduls setzt sich in gleichen Teilen aus den Noten der Teilprüfungsleistungen zusammen. Auf Antrag können die Leistungsnachweise auch in englischer Sprache oder deutscher Sprache erbracht werden.
Status	Pflichtmodul
Niveaustufe / Level	Advanced level course
Notwendige Voraussetzungen	Notwendige Voraussetzung für die Teilprüfungsleistung IT Projektmanagement ist der erfolgreiche Abschluss der Unit Moderation, Präsentation und Zeitmanagement
Empfohlene Voraussetzungen	In der Lehrveranstaltung IT Projektmanagement werden grundlegende Kenntnisse der Organisation von Unternehmen aus dem Modul Betriebswirtschaftslehre sowie Kenntnisse der verschiedenen Vorgehensmodelle bei der Softwareentwicklung aus den Modulen des Software Engineering vorausgesetzt. Das Programmierpraktikum mit Datenbanken und die Fallstudie Prozessmodellierung baut auf den Inhalten der Module aus dem Bereich der Programmierung, den Inhalten der Module aus dem Bereich des Software Engineering und dem Modul Datenbanken auf.
Lernergebnis / Kompetenzen	Die heutige Berufspraxis von Informatikern ist geprägt durch die Arbeit in Projekten. Neben der fachlichen Ausbildung in der Informatik spielen die Befähigung zur Gruppenarbeit in Projekten als auch zur Organisation und Durchführung von Projekten eine zentrale Rolle für eine erfolgreiche Tätigkeit. Die Lehrveranstaltungen des Moduls soll den Studierenden in Vorbereitung für ihre spätere Tätigkeit aber auch für die Praxisphase neben den notwendigen theoretischen Grundlagen auch konkrete Hilfestellung bei der Bewältigung projektspezifischer Aufgabenstellungen geben. Dazu werden in dem Modul neben der Vermittlung von Kenntnissen aus dem Projektmanagement die Fähigkeiten in Medien-, Präsentations- und Kommunikationstechniken gefördert sowie die in den Bereichen Konfliktmanagement und Teamarbeit, Persönlichkeitstraining und Zeitmanagement gemachte Erfahrungen aufgearbeitet. Die Verbindung von Kenntnissen und Fähigkeiten aus unterschiedlichen Bereichen wie hier der Programmierung, dem Software Engineering und der Datenbanken ist entscheidend für die spätere Berufspraxis. Mit einer Fallstudie Prozessmodellierung und dem Programmierpraktikum mit Datenbanken wird zur Befähigung Anwendungssysteme zu entwickeln und zu



	<p>beurteilen, welche Informatikprodukte zur Lösung dieser Aufgaben sinnvoll und effizient einzusetzen sind, beigetragen.</p> <p>Damit integriert dieses Modul die erworbenen Kenntnisse und Fähigkeiten und trägt damit zu allen Ausbildungszielen und Qualifikationsmerkmalen des Studiengangs bei.</p> <p>Die Studierenden sollen die Bearbeitung einer größeren realitätsnahen Aufgabe aus dem Bereich der Software-Entwicklung und Implementierung in einer Gruppe mit verteilten Zuständigkeiten kennen lernen und erste Erfahrungen in der für den Informatiker wichtigen Projektarbeit sammeln.</p> <p>Die Studierenden sollen Methoden des Projektmanagement im allgemeinen kennen und anwenden können Insbesondere sollen sie die Fähigkeit erwerben, zeit-, kosten- und ressourcenbezogene Merkmale von Projekten zu bestimmen und auszuwerten. Sie sollen speziell die Eigenarten und Risiken von DV-Projekten kennen und die allgemeinen Methoden des Projektmanagement auf solche anwenden können. Zusätzlich sollen die Studierenden erlernen,</p> <ul style="list-style-type: none"> • Ideen und Lösungsvorschläge schriftlich und mündlich zu präsentieren, • abweichende Positionen der Gesprächspartner (Auftraggeber, Kunden) zu erkennen und in eine sachgerechte Lösung zu integrieren • zur Lösung von Konflikten in kontroversen Diskussionen zielorientiert zu diskutieren und mit Kritik sachlich umzugehen und • unter Zeitdruck bzw. in Stresssituationen effizient mit der zu Verfügung stehenden Zeit umzugehen. <p>Durch die Stärkung der Teamfähigkeit sowie der methodischen und technischen Kompetenzen zur Arbeit in und mit Teams sowie zur Bewältigung von Terminstress trägt das Modul zum Erwerb außerfachlicher Kompetenzen bei.</p>
Arbeitsaufwand (h) / Gesamtworkload	300 h 90 h anteiliger Workload zum Erwerb außerfachlicher Kompetenzen
Units	<ul style="list-style-type: none"> • IT Projektmanagement • Moderation, Präsentation, Zeitmanagement • Fallstudien Prozessmodellierung • Programmierpraktikum mit Datenbanken
Häufigkeit des Angebots	Nur im Sommersemester
Anerkannte Module	Keine
Modulkoordination	N. N.
Hinweise	

21.1 Unit: IT Projektmanagement

Code	
Modulname	IT Projekte
Lernform	Vorlesung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) /	65 h



workload	
Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	<ul style="list-style-type: none"> • Projektmanagement in der Informationsverarbeitung • Formen der Projektorganisation • Phasen des Projektmanagements (Definition, Planung, Monitoring, Steuerung und Kontrolle, Berichtswesen, Dokumentation, Abschluss) • Werkzeuge des Projektmanagements, inkl. Softwareauswahl • Methoden der Aufwandsschätzung • Risikomanagement von IT-Projekten • Reife von Organisationen <ul style="list-style-type: none"> o Capability Maturity Model (CMM) / CMMI • Personalmanagement <ul style="list-style-type: none"> o Auswahl von Mitarbeitern o Motivation von Mitarbeitern o Gruppenverhalten und Gruppenführung • Das People Capability Maturity Model (P-CMM)
Literatur	Noch offen
Art und Form des Leistungsnachweises	Klausur über 90 Minuten, Zulassungsvoraussetzung ist die erfolgreiche Teilnahme an der Unit Moderation, Präsentation, Zeitmanagement
Bewertung	differenziert
Lehrende	Hefter, Wagner
Hinweise	

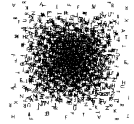
21.2 Unit: Moderation, Präsentation, Zeitmanagement	
Code	
Modulname	IT Projekte
Lernform	Übung 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	35 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	0 h
Inhalte	<ul style="list-style-type: none"> • Entwurf von schriftlichen und mündlichen Präsentationen • Rhetorische Übungen, eventuell mit Videoaufzeichnung • Diskussionen Rollenspiele zur Darstellung des Auftraggeber-/ Auftragnehmerverhältnisses
Literatur	----
Art und Form des Leistungsnachweises	Erfolgreiche Teilnahme an den Übungen, das heißt Teilnahme an mindestens 80% aller Übungen, ist Zulassungsvoraussetzung für die Teilprüfungsleistung IT Projektmanagement



Bewertung	Undifferenziert
Lehrende	Fachdozenten des Fachbereich 3
Hinweise	

21.3 Unit: Fallstudie Prozessmodellierung	
Code	
Modulname	IT Projekte
Lernform	Projekt 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	100 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	65 h
Inhalte	<ul style="list-style-type: none"> • System Engineering <ul style="list-style-type: none"> o Information Strategy Planning o Business Area Analysis <ul style="list-style-type: none"> – Datenarchitektur – Anwendungsarchitektur – Technische Infrastruktur • System Modellierung, Top-down und Bottom-up - Verfahren <ul style="list-style-type: none"> o Business System Design <ul style="list-style-type: none"> – Datenflußdiagramme – Objektorientierte Verfahren mit UML • System Simulation
Literatur	Noch offen
Art und Form des Leistungsnachweises	Erfolgreiche Durchführung und Präsentation der Fallstudie.
Bewertung	Differenziert
Lehrende	Hefter, Kratz, Wagner
Hinweise	

21.4 Unit: Programmierpraktikum mit Datenbanken	
Code	
Modulname	IT Projekte
Lernform	Projekt 2 SWS
Sprache	Deutsch bzw. Englisch je nach Modul
Arbeitsaufwand (h) / workload	100 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	65 h
Inhalte	Die bisher im Studium erlernten Kenntnisse und Erfahrungen aus dem Bereich der Programmierung, Software Engineering und Datenbanken wer-



	den innerhalb einer realitätsnahen Aufgabenstellung verarbeitet und umgesetzt. Dazu gehören: <ul style="list-style-type: none">• Vollständige Entwicklung und Implementierung einer Datenbankapplikation• Verwendung von Werkzeugen bei der Entwicklung
Literatur	Noch offen
Art und Form des Leistungsnachweises	Erfolgreiche Durchführung und Präsentation des Praktikums
Bewertung	Differenziert
Lehrende	Buhr, Hackenbracht, Lindner, Schubert
Hinweise	



Module des dritten Studienjahrs

22. Modul: Verteilte Anwendungen	
Code	
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur über 90 Minuten; auf Antrag kann die Prüfung auch in englischer Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Advanced level course
Notwendige Voraussetzungen	Erfolgreicher Abschluss der Unit Übung Verteilte Anwendungen
Empfohlene Voraussetzungen	Inhaltlich werden die Module der objektorientierten Programmierung, des Software Engineering sowie Rechnernetze und Datenbanken vorausgesetzt.
Lernergebnis / Kompetenzen	<p>Die Studierenden sollen ein Verständnis für die Konzepte und Realisierungsmöglichkeiten verteilter Anwendungen bekommen.</p> <p>Ein wesentlicher Teil der Veranstaltung besteht aus Programmierübungen, in denen verschiedenste Technologien zur Realisierung verteilter Anwendungen, die in der industriellen Praxis einen breiten Einsatz finden, benutzt werden. Dazu sind die entsprechenden Java-Komponenten auch auf einem Arbeitsplatzrechner zu installieren und zu konfigurieren. Die Studierenden werden in die Lage versetzt, unterschiedliche Anwendungspotenziale der verschiedenen Technologieansätze einzuschätzen, um selbst geeignete Anwendungen entwickeln zu können.</p> <p>Sie erlangen damit eine fundierte softwaretechnologische Problemlösungskompetenz in der Erstellung verteilter Applikationssysteme</p>
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Verteilte Anwendungen, Übung Verteilte Anwendungen
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	Keine
Modulkoordination	Klingemann
Hinweise	

22.1 Unit: Vorlesung Verteilte Anwendungen	
Code	
Modulname	Verteilte Anwendungen
Lernform	Vorlesung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	65 h



Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	<p>Neben einer Diskussion von Eigenschaften und Problemstellungen verteilter Anwendungen werden Implementierungstechnologien für die Erstellung moderner Anwendungen, die sich typischerweise durch Verknüpfung einzelner Teilsysteme entwickeln, vorgestellt. Betrachtet wird dabei eine Auswahl verschiedener Technologien aus folgenden Bereichen:</p> <ul style="list-style-type: none"> • Sockets als Basistechnologie zur Kopplung verteilter Anwendungen • Objektorientierte Middleware-Technologien wie CORBA und Java RMI • Message Queues • Web Services • Techniken zur Präsentation von Internetanfragen mit Hilfe von Java Servlets und der Scriptsprache Java Server Pages • Umsetzung von Applikationslogik mit Java Servlets • Möglichkeiten der Anbindung von Datenbanksystemen mit JDBC • Gegebenenfalls weitere, der Entwicklung des Fachgebiets entsprechende Technologien
Literatur	<ul style="list-style-type: none"> • Boger, M.: „Java in verteilten Systemen“, dpunkt.verlag, Heidelberg • Dehnhardt, W.: „Anwendungsprogrammierung mit JDBC“, Hanser-Verlag, München • Deitel, H.M., et.al.: „Advanced Java 2 Platform - How to Program“, Prentice Hall, Upper Saddle River, NJ 07458 • Eberhardt, A., et.al.: „Java-Bausteine für e-Commerce-Anwendungen“, Hanser-Verlag, München • Griffel, F.: „Componentware“, dpunkt.verlag, Heidelberg • Hofmann, J., et al.: „Programmieren mit COM und CORBA“, Hanser-Verlag • Zimmermann, J., et.al.: „Verteilte Komponenten und Datenbankanbindung“, Addison Wesley, München
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Buhr, Klingemann
Hinweise	

22.2 Unit: Übung Verteilte Anwendungen

Code	
Modulname	Verteilte Anwendungen
Lernform	Übung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	85 h
Anteil Präsenzzeit	35 h



Anteil Selbststudium	50 h
Inhalte	Ein wesentlicher Teil des Moduls besteht aus Programmierübungen, in denen die in der Vorlesung behandelten Technologien für die Realisierung einer Fallstudie praktisch zum Einsatz kommen. Dazu sind die entsprechenden Java-Komponenten auf einem NT-Arbeitsplatzrechner zu installieren und zu konfigurieren. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Verteilte Anwendungen
Art und Form des Leistungsnachweises	Leistungsnachweis: Bestehen aller zur Lehrveranstaltung angebotenen Testaten Der Leistungsnachweis ist Zulassungsvoraussetzung für die Modulprüfungsleistung.
Bewertung	Differenziert
Lehrende	Buhr, Klingemann
Hinweise	

23. Modul: Moderne Netzstrukturen

Code	
Studiengang	Informatik
Verwendbarkeit	Informatik
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Klausur über 90 Minuten Auf Antrag kann die Prüfung auch in englischer Sprache stattfinden.
Status	Pflichtmodul
Niveaustufe / Level	Specialized level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Inhaltlich werden die Informatikmodule der ersten 3 Studiensemester vorausgesetzt.
Lernergebnis / Kompetenzen	Dieses Modul bietet eine Vertiefung in einem wichtigen Spezialgebiet der Informatik. Das Modul richtet sich an Studierende, die bereits über elementare Rechnernetzkenntnisse verfügen. Ergänzend hierzu werden aktuelle Protokolle vorgestellt, die z. B. in Hochgeschwindigkeitsnetzen, mobilen Netzen oder integrierten Sprach- und Datennetzen eingesetzt werden. Die Studierenden sollen diese Konzepte und Verfahren, sowie praxisrelevante Systeme kennen lernen und verstehen
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung Moderne Netzstrukturen, Übung Moderne Netzstrukturen
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	Keine
Modulkoordination	Filip



Hinweise	
----------	--

23.1 Unit: Vorlesung Moderne Netzstrukturen	
Code	
Modulname	Moderne Netzstrukturen
Lernform	Vorlesung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	65 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	<ul style="list-style-type: none"> • Übertragung und Vermittlung (ATM, IP-Switching, MPLS) • Mobile Kommunikation: Übertragung (Funk) und Vermittlung (Mobile IP etc.) • Neue Anwendungen: Videokonferenzen, Multimediasysteme • Mehrpunktkommunikation: Dienste, Routing, zuverlässiger Transport • Anwendungsunterstützung: Session Management, Konferenz-steuerung
Literatur	<ul style="list-style-type: none"> • Jochen Schiller „Mobilkommunikation - Techniken für das allgegenwärtige Internet“, Addison-Wesley, 2000 • William Stallings, „Wireless Communications and Networks“, Prentice Hall, 2002 • James D. Solomon „Mobile IP“, Prentice Hall, 1999 • Charles E. Perking, “Ad Hoc Networking”, Addison-Wesley, 2000 • Matthew W. Gast, “802.11 Wireless Networks”, O’Reilly 2002 • Uyles Black, “ATM: Foundation for Broadband Networks”, Prentice Hall, 1993 • William Stallings, “ISDN and Broadband ISDN with Frame Relay and ATM, Prentice Hall, 1995 • George C. Sacket & Christopher Y. Metz, ATM and Multiprotocol Networking, McGraw Hill, 1996
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Filip, Kappes
Hinweise	

23.2 Unit: Übung Moderne Netzstrukturen	
Code	
Modulname	Moderne Netzstrukturen
Lernform	Übung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	85 h



Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	Die Übungen dienen dazu, dass die Studierenden die in der Vorlesung vermittelten Inhalte aufzuarbeiten und praktisch umzusetzen. Die Studierenden bekommen durch die Übungen kontinuierlich ein qualifiziertes Feedback, was ihren Lernprozess gezielt unterstützen soll.
Literatur	Siehe Unit Vorlesung Moderne Netzstrukturen
Art und Form des Leistungsnachweises	Kein Leistungsnachweis
Bewertung	Keine
Lehrende	Filip, Kappes
Hinweise	

24. Modul: Informatik Projekt

Code	
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	10 CP
Prüfungsart	Erfolgreiche Durchführung und Präsentation des Informatik Projektes Auf Antrag können die Leistungsnachweise auch in englischer Sprache erbracht werden..
Status	Pflichtmodul
Niveaustufe / Level	Advanced level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Das Modul fasst die in den Modulen der vorhergehenden Studiensemestern erarbeiteten Kenntnisse und Fähigkeiten zusammen. Von besonderer Bedeutung sind hierbei die Lehrveranstaltungen Moderation, Präsentation und Zeitmanagement und IT-Projektmanagement des Moduls IT Projekte
Lernergebnis / Kompetenzen	Das Modul Projekt soll die Absolventen befähigen, komplexe Anwendungssysteme zu entwickeln und zu beurteilen, welche Informatikprodukte zur Lösung dieser Aufgaben sinnvoll und effizient einzusetzen sind. Damit integriert dieses Modul die erworbenen Kenntnisse und Fähigkeiten und trägt damit zu allen Ausbildungszielen und Qualifikationsmerkmalen des Studiengangs bei. Die Studierenden sollen die Bearbeitung einer größeren realitätsnahen Aufgabe aus den Bereichen Software-Entwicklung und Implementierung, Entwurf und Implementierung von Rechnernetzarchitekturen, Systemmanagement und -administration in einer Gruppe mit verteilten Zuständigkeiten kennen lernen und sich im Spannungsfeld des Software Engineering zwischen Stand der Wissenschaft und Umsetzung in der Praxis orientieren können. Das Modul trägt durch die Stärkung der Teamfähigkeit zum Erwerb außerfachlicher Kompetenzen bei.
Arbeitsaufwand (h) / Gesamtworkload	300 h 60 h anteiliger Workload zum Erwerb außerfachlicher Kompetenzen



Units	Projekt
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	Keine
Modulkoordination	N. N.
Hinweise	

24.1 Unit: Projekt

Code	
Modulname	Informatik Projekt
Lernform	Projekt 4 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	300 h
Anteil Präsenzzeit	70 h
Anteil Selbststudium	230 h
Inhalte	<p>In dem Modul werden die im Studium erlernten Kenntnisse und Erfahrungen, die am Stand der Wissenschaft orientiert sind und die in der Praxisphase gemachten Erfahrungen verarbeitet werden.</p> <ul style="list-style-type: none"> • Vollständige Bearbeitung einer größeren realitätsnahen Aufgabe aus dem Bereich der Software-Entwicklung und Implementierung nach den Methoden des Software Engineering. • Verwendung von Werkzeugen bei der Entwicklung • Das Thema des Projekts umfasst interdisziplinär mehrere Kernfächer der Informatik
Literatur	Abhängig vom Projekt
Art und Form des Leistungsnachweises	Siehe Modulbeschreibung
Bewertung	
Lehrende	Informatik Dozenten
Hinweise	

25. Modul: Aktuelle Themen der Informatik

Code	
Studiengang	Informatik
Verwendbarkeit	Verwendbar in anderen Informatik-Studiengängen
Dauer	1 Semester
Credits	5 CP
Prüfungsart	<p>In dem Modul sind zwei Teilprüfungsleistungen in den Units des Moduls zu erbringen.. Die Note des Moduls setzt sich zu gleichen Teilen aus den Noten der Teilprüfungsleistungen zusammen.</p> <p>Auf Antrag können die Leistungsnachweise auch in englischer Sprache erbracht werden.</p>
Status	Wahlpflichtmodul



Niveaustufe / Level	Specialized level course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Die Lehrveranstaltungen sollen auf Fächern des Pflichtbereichs aufbauen.
Lernergebnis / Kompetenzen	Zusätzlich zu den Pflichtveranstaltungen wird die individuelle Spezialisierung der Studierenden in einem Wahlpflichtbereich durch das Angebot von Spezialveranstaltungen unterstützt. Ein wesentliches Lernziel ist das eigenständige wissenschaftliche Arbeiten in einem Thema der Informatik.
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Vorlesung, Seminar
Häufigkeit des Angebots	Nur im Wintersemester
Anerkannte Module	keine
Modulkoordination	Studiengangsleiter Informatik
Hinweise	Es ist jeweils eine Vorlesung sowie ein Seminar aus dem Katalog der Wahlpflichtfächer zu wählen.

25.1 Unit: Vorlesung

Code	
Modulname	Wahlpflicht
Lernform	Vorlesung 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	65 h
Anteil Präsenzzeit	35 h
Anteil Selbststudium	30 h
Inhalte	Aktuelle Themen der Informatik oder Vertiefungen von Themen des Pflichtprogramms
Literatur	Abhängig von der Thematik
Art und Form des Leistungsnachweises	Klausur 90 Minuten
Bewertung	differenziert
Lehrende	
Hinweise	

25.2 Unit: Seminar

Code	
Modulname	Wahlpflicht
Lernform	Seminar 2 SWS
Sprache	Deutsch
Arbeitsaufwand (h) / workload	85 h



Anteil Präsenzzeit	35 h
Anteil Selbststudium	50 h
Inhalte	Aktuelle Themen der Informatik oder Vertiefungen von Themen des Pflichtprogramms
Literatur	Abhängig von der Thematik
Art und Form des Leistungsnachweises	Schriftliche Ausarbeitung und Präsentation
Bewertung	Differenziert
Lehrende	
Hinweise	

26. Modul: Studium Generale	
Code	
Studiengang	
Verwendbarkeit	Das Modul kann im Rahmen des Studiums Generale in allen Studiengängen Verwendung finden.
Dauer	1 Semester
Credits	5 CP
Prüfungsart	Das Modul wird mit einer Prüfungsleistung abgeschlossen. Gemäß § 10 der „Allgemeinen Bestimmungen für Prüfungsordnungen...“ können eine mündliche Prüfung, schriftliche Prüfung oder Projektarbeiten durchgeführt werden. Die Art der Prüfungsleistung ist abhängig von der jeweiligen Ausgestaltung des Moduls „Studium Generale“. Die jeweilige Prüfungsform ergibt aus dem gewähltem Modul.
Status	Wahlpflichtmodul
Niveaustufe / Level	Specialized Level Course
Notwendige Voraussetzungen	Keine
Empfohlene Voraussetzungen	Keine; empfohlene Voraussetzungen: 60 ECTS im Fachstudium
Lernergebnis / Kompetenzen	<p>Das Modul zum „Studium Generale“ bildet das Profilvermerkmal der Interdisziplinarität der FH FFM auf der Ebene der einzelnen Studiengänge ab. Es handelt sich um ein Modul, bei dem aus den vier bzw. aus mindestens drei Fachbereichen zu einem Querschnittsthema fachliche Beiträge integrativ verknüpft und den Studierenden aller Fachbereiche zum Kompetenzerwerb verpflichtend angeboten werden.</p> <p>Die Studierenden</p> <ul style="list-style-type: none"> • sind zu interdisziplinärem Denken und kooperativem Handeln fähig; • überwinden die Begrenztheit ihrer fachspezifischen Denkweisen (Theorien und Methoden); • sind in der Lage, naturwissenschaftliche und technische, wirtschaftliche und rechtliche, kulturelle, soziale und persönliche Aspekte am Beispiel eines Querschnitt-Themas zu erkennen, diese gegeneinander abzuwägen und ganzheitlich zu reflektieren; • können Zusammenhänge ihres Fachs im Raum unterschiedlicher wissen-



	<p>schaftlicher Disziplinen sowie gesellschaftlicher Interessen verständlich machen (kommunizieren, präsentieren und argumentieren); reflektieren die Wirkungen und Folgen ihrer beruflichen und gesellschaftlichen Tätigkeit und können daraus Konsequenzen für ihr eigenes Handeln ableiten.</p> <p>Durch die Stärkung der Teamfähigkeit insbesondere in Teams mit unterschiedlichem fachlichem Hintergrund sowie Einblicke in andere Fächer und Fachkulturen trägt das Modul zum Erwerb außerfachlicher Kompetenzen bei.</p>
Inhalte	Ein Querschnittsthema unter Beteiligung von mindestens drei Fachbereichen, z.B. Demografischer Wandel, Energie, Ethik, Fundraising, Gender Mainstreaming, Gläserner Mensch, Globalisierung, Kommunikation und Medien, Krisenintervention und Versagensprävention, Managing Diversity, Mobilität, Musik, Organisationsentwicklung, Wirtschaftspolitik, Wissenschaftskonzepte, ...
Literatur	
Arbeitsaufwand (h) / Gesamtworkload	150 h
Units	Die Units ergeben sich aus dem entsprechenden Modul.
Häufigkeit des Angebots	Hängt von dem entsprechenden Modul ab
Anerkannte Module	
Modulkoordination	N. N.
Hinweise	

27. Modul: Praxisphase	
Code	
Studiengang	Informatik
Verwendbarkeit	Informatik
Dauer	1 Semester
Credits	18 CP
Prüfungsart	<p>Bericht und Vortrag sowie Teilnahme an 80% aller Seminartermine. Für versäumte Seminartermine ist eine Entschuldigung vorzulegen (z.B. ärztliches Attest oder Bescheinigung des Praxisbetriebes über Schulungsteilnahme oder Messebesuch). Wird die Praxisphase im Ausland absolviert, sind zwei Berichte (in der Mitte und am Ende der Praxisphase) zu erstellen, die Teilnahme am Seminar kann entfallen.</p> <p>Für Bericht und Vortrag zusammen wird eine Note erteilt unter der Voraussetzung, dass die Anwesenheitspflicht erfüllt wurde. Wird die Praxisphase im Ausland absolviert, wird die Note für beide Berichte zusammen erteilt.</p> <p>Auf Antrag können die Leistungsnachweise auch in englischer Sprache erbracht werden.</p>
Status	Pflichtmodul
Niveaustufe / Level	Advanced level course
Notwendige Voraus-	Erfolgreicher Abschlusse aller Module der ersten 4 Studiensemester; im



setzungen	<p>Einzelnen sind dies die Module Mathematische Grundlagen – Algebra, Grundlagen der Informatik und maschinennahe Programmierung, Einführung in die Programmierung mit C, Betriebswirtschaftslehre, Englisch, Diskrete Mathematik, Digitaltechnik, Theoretische Grundlagen der Informatik, Algorithmen und Datenstrukturen, Objektorientierte Programmierung Grundlagen und Konzepte, Software Engineering – Analyse, Statistik, Objektorientierte Programmierung Vertiefung – Java, Datenbanken, Rechnernetze, Betriebssysteme, Numerische Mathematik, Software Engineering – Design, Echtzeitsysteme, Datenschutz / IT-Security, IT-Projekte. Außerdem muss</p> <ul style="list-style-type: none"> • eine Bescheinigung der Praxisfirma und des Praxis-Beauftragten über die Erfüllung der Praxisaufgaben vorliegen und • der Bericht von der Firma freigegeben sein.
Empfohlene Voraussetzungen	Alle Module der ersten fünf Studiensemester
Lernergebnis / Kompetenzen	<p>Die Praxisphase unterstützt die Orientierung der Studierenden im angestrebten Berufsfeld und erleichtert die Aufnahme einer späteren Berufstätigkeit. Die hier gemachten Erfahrungen werden in einem begleitenden Seminar nachgearbeitet und vertieft. Mit den aufbereiteten Erkenntnissen aus der beruflichen Praxis lassen sich Inhalte von Vorlesungen besser verarbeiten und hinsichtlich einer künftigen beruflichen Tätigkeit besser einordnen. Damit dient das Modul der Erfolgskontrolle und der Motivation für die Bachelorarbeit. Neben der fachlichen Arbeit erhalten die Studierenden gleichzeitig Einblicke in betriebliche Abläufe und Organisationen. Selbständiges, verantwortungsbewusstes Handeln wird gefördert.</p> <p>Die Orientierung im angestrebten Berufsfeld, die Förderung von selbständigem und verantwortungsbewusstem Handeln in Kooperation mit Anderen trägt zum Erwerb der außerfachlichen Kompetenzen bei.</p> <p>Ein wesentliches Lernziel ist die Anwendung der bisher im Studium erworbenen Kenntnisse und Fähigkeiten in Form von verantwortlichem eigenständigem Arbeiten in Kooperation mit anderen. Im Einzelnen sind Lernziele:</p> <ul style="list-style-type: none"> • Fähigkeit zu verantwortlicher Arbeit in Kooperation mit anderen • Fähigkeit zur Beurteilung von fremden Software-Systemen • Einblick in wichtige Anwendungsfelder der Informatik • Verständnis der Bedeutung der IT für das Unternehmen und die Gesellschaft • Die Fähigkeit, einen Vortrag zur beruflichen Tätigkeit selbstständig zu erarbeiten und diesen Vortrag unter Nutzung moderner Präsentationstechniken in einem vorgegebenen Zeitrahmen zu halten. • Die Fähigkeit, einen mehrseitigen Bericht in ansprechender Form zu verfassen.
Arbeitsaufwand (h) / Gesamtworkload	540 h 240 h anteiliger Workload zum Erwerb außerfachlicher Kompetenzen
Units	Seminar zur Praxisphase, Betreutes Praxisprojekt
Häufigkeit des Angebots	Jährlich im Sommersemester. Beginn zwischen 1.3. und 1.4. des Jahres.



bots	
Anerkannte Module	keine
Modulkoordination	Güsmann
Hinweise	Zur Vorbereitung und zur Organisation der Praxisphase erfolgt im Vorse- mester eine einführende Veranstaltung, deren Termin jeweils per Aushang mitgeteilt wird.

27.1 Unit: Seminar zur Praxisphase

Code	
Modulname	Praxisphase
Lernform	Seminar
Sprache	Deutsch
Arbeitsaufwand (h) / workload	40 h
Anteil Präsenzzeit	20 h
Anteil Selbststudium	20 h
Inhalte	Zeitgemäße Präsentation der Praxisprojekte in einem 20-minütigen Vortrag mit <ul style="list-style-type: none"> • Diskussion zum eigenen Vortrag und Beteiligung an der Diskussion zu anderen Vorträgen. • Layout von Präsentationsseiten (Folien/Laptop/Beamer) • Erstellung eines 15-30-seitigen Berichtes zum Vortrag in optisch ansprechender Form mit korrekter Rechtschreibung. Der Bericht muss bis zum Tag des Vortrages vom Praxis-Betrieb durch Stempel und Unterschrift freigegeben worden sein und dem Seminarleiter vor Beginn des Vortrages vorgelegt worden sein.
Literatur	Keine
Art und Form des Leistungsnachweises	Siehe Modul
Bewertung	Siehe Modul
Lehrende	Güsmann, Wagner, Hackenbracht
Hinweise	Während der Vorlesungszeiten findet freitags das Seminar zur Praxisphase statt. Jeder Vortrag dauert 20 Minuten mit anschließender Diskussion.

27.2 Unit: Betreutes Praxisprojekt

Code	
Modulname	Praxisphase
Lernform	Projekt außerhalb der Hochschule
Sprache	Deutsch, Englisch oder eine Landessprache beim Praxisprojekt im Ausland
Arbeitsaufwand (h) / workload	500 h
Anteil Präsenzzeit	----
Anteil Selbststudium	----
Inhalte	Qualifizierte Mitarbeit an einem oder an mehreren kleinen Projekten aus den Gebieten



	<ul style="list-style-type: none"> • Systemanalyse • Projektierung • betrieblicher Einsatz von Standardsoftware • Anwendungsprogrammierung • Systemprogrammierung
Literatur	Die Literatur wird von der Firma gestellt, in der das betreute Praxisprojekt absolviert wird. Ergänzende Literaturquellen sind eigenständig zu beschaffen, gegebenenfalls nach Beratung durch die Dozenten.
Art und Form des Leistungsnachweises	Bescheinigung der Praxisfirma und des Praxis-Beauftragten über die Erfüllung der Praxisaufgaben
Bewertung	Undifferenziert
Lehrende	----
Hinweise	Das Praxisprojekt umfasst 14 Wochen a 4,5 Tagen. Die Teilnahme an dem Seminar zur Praxisphase muss durch die Firma ermöglicht werden.

28. Modul: Bachelorarbeit

Code	
Studiengang	Informatik
Verwendbarkeit	Informatik
Dauer	1 Semester
Credits	12 CP
Prüfungsart	Schriftliche Ausarbeitung mit anschließendem Kolloquium Die Note des Moduls ergibt sich zu 70% aus der Note der schriftlichen Ausarbeitung und zu 30% aus der Note des Kolloquiums. Auf Antrag kann die Arbeit auch in englischer Sprache erstellt werden und das Kolloquium in englischer Sprache durchgeführt werden..
Status	Pflichtmodul
Niveaustufe / Level	----
Notwendige Voraussetzungen	Nachweis des Beginns des Moduls Praxisphase in Form eines unterzeichneten Ausbildungsvertrages sowie erfolgreicher Abschluss aller Module der ersten 5 Studiensemester.
Empfohlene Voraussetzungen	Alle Module der ersten 5 Studiensemester
Lernergebnis / Kompetenzen	<p>Mit der Bachelorarbeit sollen die Studierenden zeigen, dass sie in der Lage sind, eine komplexe, praxisbezogene Aufgabe mit wissenschaftlichen Methoden des Fachs zu bearbeiten. Mit schriftlichen Ausarbeitungen sollen sie die Ergebnisse übersichtlich darstellen, wissenschaftlich fundiert bewerten und Entscheidungen verständlich begründen . Im abschließenden Kolloquium soll sie ihre Arbeit gegenüber fachlicher Kritik vertreten können.</p> <p>Nach erfolgreichem Abschluss des Moduls sollen die Studierenden</p> <ul style="list-style-type: none"> • komplexe, praxisbezogene Aufgaben mit wissenschaftlichen Methoden des Fachs bearbeiten • Wissenschaftlich fundierte, schriftliche Ausarbeitungen erstellen können.



	<ul style="list-style-type: none"> eigene Ideen und Ergebnisse gegenüber fachlicher Kritik öffentlich vertreten können.
Inhalte	Die Bachelorarbeit ist eine wissenschaftliche Arbeit über ein abgeschlossenes Thema. Die Arbeit soll sich bevorzugt auf die Ergebnisse der Praxisphase gründen, die in Industrieunternehmen, an ausländischen Partnerhochschulen oder in wissenschaftlichen Einrichtungen durchgeführt wurde
Literatur	----
Arbeitsaufwand (h) / Gesamtworkload	360 h
Units	----
Häufigkeit des Angebots	Nur im Sommersemester
Anerkannte Module	Keine
Modulkoordination	----
Hinweise	Das Kolloquium findet am Ende der Arbeit mit den Referenten statt. Im Kolloquium zur Bachelorarbeit soll die oder der Studierende ihre oder seine Bachelorarbeit gegenüber kritischen Fragen verteidigen.



Zusatzqualifikation „Betrieblicher Datenschutz“

Studenten des Studiengangs Informatik und des Studiengangs Wirtschaftsinformatik können eine Zusatzqualifikation "Betrieblicher Datenschutz" erwerben. Hier soll das Rüstzeug zur selbständigen, effizienten und erfolgreichen Durchführung der Aufgaben eines betrieblichen Datenschutzbeauftragten vermittelt werden. In Zusammenarbeit mit den Fachhochschulen in Kiel und München wird ein einheitliches Zertifikat verliehen.

Mit dieser Zusatzqualifikation werden

- die Fähigkeit zu verantwortlicher Arbeit in Kooperation mit anderen,
- die Fähigkeit zur Beurteilung von fremden Software-Systemen,
- der Einblick in wichtige Anwendungsfelder der Informatik und
- das Verständnis der Bedeutung der IT für das Unternehmen und die Gesellschaft gefördert.

Lernziele der Zusatzqualifikation sind:

- Bearbeitung rechtlicher Fragen in Zusammenhang mit dem allgemeinen Persönlichkeitsschutz, dem Grundrecht auf Datenschutz, den gesetzlich geschützten Sondergeheimnissen
- Vertiefte Kenntnisse der technischen Grundlagen und Probleme der Informationssicherheit
- Vermittlung von Kenntnissen der wichtigsten Datenschutzgesetze und -normen
- Strategien zur Umsetzung des Datenschutzes im betrieblichen Umfeld
- Intensive Kenntnisse und praktische Erfahrungen bezüglich möglicher Massnahmen zum Datenschutz und zur Datensicherung
- Umgang mit Gefährdungen im Bereich Datenschutz/IT-Sicherheit, Analyse von Risikofaktoren

Die Zusatzqualifikation besteht aus innerhalb des Studiengangs Informatik angebotenen Modulen sowie zusätzlich angebotenen Lehrveranstaltungen, die teilweise auch im Studiengang Wirtschaftsinformatik angeboten werden. Die Integration in den Studiengang Informatik wird durch folgende Tabelle deutlich:

Lehrveranstaltung	Umfang, Turnus, Workload	Einbindung in das Studienprogramm
Datenschutz	2 SWS, SS, 75 h	Unit des Moduls Datenschutz / IT Security des 4. Studiensemester
Datenschutz Vertiefung	2 SWS, WS, 60 h	Zusätzlich im 5. Studiensemester gemeinsam mit dem Studiengang Wirtschaftsinformatik
Seminar Aktuelle Datenschutzprobleme	2 SWS, SS, 90 h	Blockseminar, zusätzlich im 6. Studiensemester gemeinsam mit dem Studiengang Wirtschaftsinformatik
Wirtschaftsprivatrecht: Einführung	WS, 2 SWS, 60 h	Zusätzlich im 3. Studiensemester Besuch des Moduls Recht im Studiengang Wirt-



Vertiefung:	SS, 2 SWS, 60 h	Wirtschaftsinformatik Zusätzlich im 4. Studiensemester gemeinsam mit dem Studiengang Wirtschaftsinformatik
IT-Security	2 SWS, SS, 75 h	Teil des Moduls Datenschutz / IT Security des 4. Studiensemester
Datenschutz-Praktikum	1 x im Jahr	Geeignetes Projekt im 5. Studiensemester

Die Zusatzqualifikation wird erworben durch die erfolgreichen Leistungsnachweise in diesen Lehrveranstaltungen. Dabei kann die Lehrveranstaltung Datenschutz Vertiefung als Vorlesung des Moduls Wahlpflicht anerkannt werden. Das Datenschutz-Praktikum wird durch die erfolgreiche Teilnahme an einem Projekt, welches Fragestellungen des Datenschutzes und der Datensicherheit zum Thema hat, nachgewiesen. Dieses wird jeweils durch den Prüfungsausschuss benannt. Zum Verständnis der Probleme des Datenschutzes und der Datensicherheit im Rahmen der Lehrveranstaltungen der Zusatzqualifikation ist ein strukturelles Wissen der Funktions- und Arbeitsweise moderner IT-Systeme erforderlich. Dies wird in den Informatik-Lehrveranstaltungen des Studiengangs vermittelt.

Die Module / Lehrveranstaltungen

- Datenschutz Vertiefung
- Aktuelle Datenschutzprobleme
- Wirtschaftsprivatrecht Einführung → Modul Recht (Wirtschaftsinformatik)
- Wirtschaftsprivatrecht Vertiefung

werden von den Bachelor-Studiengängen Informatik und Wirtschaftsinformatik gemeinsam genutzt. Ihre Beschreibungen sind in dem ECTS-Handbuch des Studiengangs Wirtschaftsinformatik zu finden.

Bei erfolgreichem Abschluss der Zusatzqualifikation wird ein Zertifikat vergeben. Näheres regelt die Satzung.