

## Inhalt

<i>Gebietszuordnung</i> .....	3
<i>Vertiefung Bildverarbeitung</i> .....	8
<i>Vertiefung Computergraphik und Visualisierung</i> .....	8
<i>Vertiefung Information Systems Engineering</i> .....	9
<i>Vertiefung Optimierung</i> .....	10
<i>Vertiefung Theoretische Informatik</i> .....	11
<i>Vertiefung Wissenschaftliches Rechnen</i> .....	12
<i>Module aus der Informatik</i> .....	14
Algorithmische Geometrie (IAGM) .....	14
Computergraphik 1 (ICG1) .....	15
Computergraphik 2 (ICG2) .....	16
Compilerbau (ICOM) .....	17
Computerspiele (ICS).....	18
Foundations of Cryptography and Network Security 1 (ICNS1) .....	19
Foundations of Cryptography and Network Security 2 (ICNS2) .....	20
Fuddokumentation mittels optischer 3D-Scanner (I3DOK) .....	22
Data Mining - Algorithmen und parallele Verfahren (IDM).....	23
Data Warehouses (IDW).....	25
Effiziente Algorithmen 1 (IEA1).....	26
Effiziente Algorithmen 2 (IEA2).....	27
Software-Praktikum für Fortgeschrittene (IFM) .....	29
Formale Sprachen und Automatentheorie (IFSA) .....	30
Flow and Transport in Terrestrial Systems (IFTTS) .....	31
Grundlagen der wissenschaftlichen Visualisierung (IGWV) .....	32
Inverse Problems (IIP) .....	34
Knowledge Discovery in Databases (IKDD) .....	35
Künstliche Intelligenz (IKI) .....	36
Konsistenzmodelle für parallele und verteilte Systeme (IKPV).....	37
Logik in der Informatik (ILI).....	38
Masterarbeit (IMa) .....	39
Management und Analyse von Datenströmen (IMADS).....	40
Mixed-Integer Programming and Combinatorial Optimization (IMIP) .....	41
Neuroscientific Computing (INSC) .....	42
Numerik von Transportprozessen in porösen Medien (INTPM) .....	44
Objekterkennung und automatisches Bildverstehen (IOAB).....	45
Objektorientiertes Programmieren im Wissenschaftlichen Rechnen (IOPWR).....	46
Paralleles Höchstleistungsrechnen (IPHR).....	47
Parallele Lösung großer Gleichungssysteme (IPLGG) .....	48
IT-Projektmanagement (IPM).....	50
Randomisierte Algorithmen (IRA).....	51
Räumliche Datenbanken (IRDB) .....	52
Seminar (IS).....	54
Simulationswerkzeuge (ISIMW).....	54
Software Evolution (ISWEvol).....	56
Software Ökonomie (ISWÖk).....	57
Wissensmanagement und Entscheidungen im Software Engineering (ISWKM).....	59
Qualitätsmanagement (ISWQM) .....	61

Requirements Engineering (ISWRE) .....	62
Visualisierung im Bereich Cultural Heritage (IVCH) .....	63
Verteilte Datenbanken und Informationssysteme (IVDB) .....	63
Verteilte Systeme I (IVS1) .....	65
Visualisierung in Natur- und Technikwissenschaften (IVNT) .....	67
Volumenvisualisierung (IVV) .....	68
Wissenschaftliches Arbeiten (IWA) .....	69
<i>Module, die auch im Bachelor/Master Mathematik angeboten werden:</i> .....	71
Mathematische Logik (MB9) .....	71
Wahrscheinlichkeitstheorie (MC4).....	72
Numerik (MD1) .....	73
Statistik (MD2).....	74
Lineare Optimierung (MD3).....	74
Nichtlineare Optimierung (MD4) .....	75
Wissenschaftliches Rechnen (MD5) .....	76
Computeralgebra I (MG19) .....	77
Computeralgebra II (MG20) .....	78
Numerische Lineare Algebra (MH5) .....	79
Numerik partieller Differentialgleichungen (MH7) .....	80
Numerische Optimierung bei Differentialgleichungen (MH8) .....	81
Statistik II (MH12) .....	82
Wahrscheinlichkeitstheorie II (MH13) .....	83
Berechenbarkeit und Komplexität I (MH14).....	84
Berechenbarkeit und Komplexität II (MH15).....	85
Algorithmische Optimierung I (MH16).....	86
Algorithmische Optimierung II (MH17).....	86
Mustererkennung (MH18).....	87
Bildverarbeitung (MH19) .....	88
Modellierung und Optimierung in Robotik und Biomechanik (MH24) .....	89
<i>Module, die im Master Technische Informatik angeboten werden:</i> .....	91
<i>Module, die auch im Bachelor/ Master Physik angeboten werden</i> .....	91
Physics of Imaging (MWInf5) .....	91
Image Processing (MWInf6).....	92
Pattern Recognition (MWInf7).....	93
Introduction to Image Processing on the GPU (IGPU).....	94
Fast Parallel Implementations of Image Labeling Problems (IMLP) .....	95
<i>Module, die auch im Bachelor/ Master Biologie angeboten werden</i> .....	97
Model-based segmentation of biomedical images (Modellbasierte Segmentierung Biomedizinischer Bilder, IMSBI) .....	97
<i>Module im Anwendungsgebiet</i> .....	98
Anwendungsgebiet (IAG).....	98

Im Folgenden sind die Module des Masterstudiengangs Angewandte Informatik beschrieben. Es können (aber müssen nicht) Vertiefungen gewählt werden, dabei sind die nachfolgenden Anmerkungen zu beachten.

Wie in der Prüfungsordnung beschrieben, sind bei der Wahl der Module Gebiet abzudecken. Die Zuordnung der Module zu den Gebieten ist nachfolgend beschrieben..

### **Gebietszuordnung**

Entsprechend der in der Prüfungsordnung genannten Vorgaben ist eine Auswahl aus den folgenden Gebieten abzudecken.

Bildverarbeitung (BV)

Computergraphik und Visualisierung (CGV)

Datenbanksysteme (DB)

Diskrete und Kombinatorische Optimierung (Opt)

Parallele und Verteilte Systeme (PVS)

Software Engineering (SWE)

Technische Informatik (Tecl)

Theoretische Informatik (Theol)

Wissenschaftliches Rechnen (WR)

Die nachfolgende Tabelle definiert die Gebietszuordnung der im Modulhandbuch beschriebenen Module. Alle nicht genannten Module sind keinem speziellen Gebiet zugeordnet.

Modul	B V	CG V	DB	Opt	PVS	SWE	Tecl	Theol	WR
Algorithmische Geometrie (IAGM)		x							
Computergraphik 1 (ICG1)		x							
Computergraphik 2 (ICG2)		x							
Computerspiele (ICS)		x							
Foundations of Cryptography and Network Security 1 und 2 (ICNS1, ICNS2)					x				
Funddokumentation (I3Doc)									
Datenbanken 2 (IDB2)			x						
Data Mining (IDM)					x				
Data Ware-houses (IDW)			x						
Effiziente Algorithmen 1 (IEA1)				x					

Effiziente Algorithmen 2 (IEA2)				x					
Formale Sprachen und Automatentheorie (IFSA)								x	
Flow and Transport in Terrestrial Systems (IFTTS)									x
Grundlagen der wissenschaftlichen Visualisierung (IGWV)		x							
Inverse Problems (IIP)		x							
Knowledge Discovery in Databases (IKDD)			x						
Künstliche Intelligenz (IKI)		x							
Konsistenzmodelle für parallel und verteilte Systeme (IKPV)					x				
Logik in der Informatik (ILI)								x	
Management und Analyse von Datenströmen (IMADS)			x						
Modellierung in Robotik (MH24)									x
Neuroscientific Computing (INSC)									x
Mixed-Integer Programming and Combinatorial Optimization (IMIP)				x					
Netzwerk-analyse									x
Numerik von Transportprozessen in porösen Medien (INTPM)									x
Objektorientiertes Programmieren im Wissenschaftlichen Rechnen (IOPWR)									x
Paralleles Höchstleistungsrechn									x

en (IPHR)									
Parallele Lösung großer Gleichungssysteme (IPLGG)									x
IT-Projektmanagement (IPM)						x			
Randomisierte Algorithmen (IRA)								x	
Räumliche Datenbanken (IRDB)			x						
Simulationswerkzeuge (ISIMW)									x
Sicherheit in Rechnersystemen (ISIR)					x				
Software-Evolution (ISWEvolv)						x			
Wissensmanagement und Entscheidungen im Software Engineering (ISWKM)						x			
Software-Ökonomie (ISWÖk)						x			
Qualitätsmanagement (ISWQM)						x			
Requirements Engineering (ISWRE)						x			
Visualisierung im Bereich Cultural Heritage (IVCH)		x							
Verteilte Datenbanken und Informationssysteme (IVDB)			x						
Visualisierung in Natur- und Technikwissenschaften (IVNT)		x							
Verteilte Systeme I (IVS1)					x				
Volume Visualization (IVV)		x							
Mathematische Logik (MB9)								x	
Numerik (MD1)									x
Lineare Optimierung				x					

(MD3)									
Nichtlineare Optimierung (MD4)				x					
Wissenschaftliches Rechnen (MD5)									x
Numerische Lineare Algebra (MH5)									x
Numerik Partieller Differentialgleichungen (MH7)									x
Numerische Optimierung bei Differentialgleichungen (MH8)				x					
Numerische Methoden der Kontinuumsmechanik (MH9)				x					
Spezielle Themen der Numerik (MH10)				x					
Spezielle Themen der Optimierung (MH11)				x					
Berechenbarkeit und Komplexität I (MH14)								x	
Berechenbarkeit und Komplexität II (MH15)								x	
Algorithmische Optimierung I (MH16)				x					
Algorithmische Optimierung II (MH17)				x					
Physics of Image Processing (MWInf5)	x								
Bildverarbeitung (MH19 oder MWInf6)	x								
Statistische Mustererkennung (MH18 oder MWInf7)	x								
Objekterkennung und automatisches Bildverstehen (IOAB)	x								
Modellbasierte Segmentierung biomedizinischer Bilder (IMSBI)	x								

Introduction to Image Processing on the GPU (IGPU)	x								
Fast Parallel Implementations of Image Labeling Problems (IMLP)	x								
Alle Grundlagen- und Vertiefungs-Module aus dem Master Technische Informatik							x		

## **Vertiefung Bildverarbeitung**

Die Vertiefung wird gestaltet vom Heidelberger Zentrum für Bildverarbeitung (Heidelberg Collaboratory Image Processing). Sie kann im Master gewählt werden. Sie umfasst 2 Seminare, Wahlpflichtveranstaltungen und die Masterarbeit. Es sind 2 Seminare und vertiefende Vorlesungen aus dem Gebiet der Bildverarbeitung zu wählen und die Masterarbeit in diesem Gebiet anzufertigen. Weitere Module in der Informatik sowie das Anwendungsgebiet können gemäß der Prüfungsordnung frei gewählt werden.

Die Veranstaltungen aus dem Gebiet der Bildverarbeitung sind

Physics of Image Processing (MWInf5)  
Bildverarbeitung (MWInf6 oder MH19)  
Statistische Mustererkennung (MWInf7 oder MH18)  
Objekterkennung und automatisches Bildverstehen (IOAB)  
Multidimensionale Signalanalyse (TIMDS)

Dabei sind die Module MWInf6 und MWInf7 mehr auf die Physik ausgerichtet und die Module MH18 und MH19 mehr auf die Mathematik. Sie zählen jeweils 8 LP.

Veranstaltung	Semester	LP
Wiss. Arbeiten	1	2
Seminar – Bildverarbeitung	1-3	4
Seminar – Bildverarbeitung	1-3	4
Bildverarbeitung (MWInf6 oder MH19)	1-3	8
Statistische Mustererkennung (MWInf7 oder MH18)	1-3	8
Weitere Module aus dem Gebiet der Bildverarbeitung	1-3	20
Weitere Module aus mindestens 2 anderen Gebieten	1-3	26
Masterarbeit Bildverarbeitung	4	30
Anwendungsgebiet	1-3	18
LP Summe		120

## **Vertiefung Computergraphik und Visualisierung**

Diese Vertiefung befähigt zur Entwicklung von Algorithmen und Anwendungsprogrammen für die visuelle Datenverarbeitung und -analyse. Hierzu gehören Kenntnisse im Bereich Mensch-Maschine-Interaktion, Computergraphik, Datenanalyse und wissenschaftliche Visualisierung. Die Vertiefung kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst mindestens ein Seminar, ein Praktikum, die Masterarbeit und zusätzliche Wahlpflichtveranstaltungen. Die Module sind aus dem Lehrgebiet Computergraphik und Visualisierung (CGV) zu wählen. Weitere Veranstaltungen sollte aus zwei ergänzenden Lehrgebieten gewählt werden.

Vertiefende Module aus dem Gebiet CGV sind  
IAGM (4 LP): Algorithmische Geometrie  
ICG1 (6 LP): Computergraphik 1



ICG2 (6 LP): Computergraphik 2  
 ICS (8 LP): Computerspiele  
 IGWV (8 LP): Grundlagen der wissenschaftlichen Visualisierung  
 IKI (6 LP): Künstliche Intelligenz  
 IVV (8 LP): Volume Visualization  
 IVNT (4 LP): Visualisierung in Natur- und Technikwissenschaften  
 IVCH (2 LP): Visualisierung im Bereich Cultural Heritage

Für das ergänzende Lehrgebiet bieten folgende Bereiche eine gute Erweiterung.

- Bildverarbeitung
- Datenbanken
- Optimierung
- Software Engineering
- Wissenschaftliches Rechnen

Weitere Bereiche können auf Antrag genehmigt werden.

Veranstaltung	Semester	LP
Wiss. Arbeiten	1	2
Seminar – Lehrgebiet CGV	1-3	4
Seminar – CGV oder ergänzendes Lehrgebiet	1-3	4
Fortgeschrittenen-Praktikum CGV	1-3	8
Vertiefende Module aus dem Bereich CGV	1-3	20
Weitere Module aus mindestens 2 weiteren Gebieten und CGV	1-3	34
Masterarbeit CGV	4	30
Anwendungsgebiet	1-3	18
LP Summe		120

Bei Belegung der Vertiefung im Bachelor und Master müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

### ***Vertiefung Information Systems Engineering***

Die Vertiefung kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst die Seminare, Praktika, Bachelor- und Masterarbeiten und die Wahlpflichtveranstaltungen.

Dabei sind die Module zu gleichen Teilen auf die beiden Lehrgebiete Datenbanken (DB) und Software Engineering (SWE) verteilt. Kernstück ist ein die beiden Lehrgebiete übergreifendes Praktikum (genannt ISE-Projekt) im Umfang von 16 LP, in dem im Team für externe Kunden ein Informationssystem mit ingenieurmäßigen Methoden unter Nutzung modernster Technologie entwickelt wird. Das ISE-Projekt wird immer im Wintersemester angeboten.

Vertiefende Module aus dem Gebiet SWE (aus dem Modulhandbuch des Masters)  
 ISWRE (8LP): Requirements Engineering

ISWQM (8LP): Qualitätsmanagement

Vertiefende Module aus dem Gebiet DB (aus dem Modulhandbuch des Masters)

IDW (4LP) DataWarehouses

IRDB (8LP) Räumliche Datenbanken

IVDB (8 LP) Verteilte Datenbanken und Informationssysteme

IKDD (8LP) Knowledge Discovery in Databases

Weitere angebotene Module können auf Antrag hinzugenommen werden

DB: Management und Analyse von Datenströmen (4LP)

SWE: IT-Projektmanagement (2LP)

SWE: Software-Ökonomie (3LP)

SWE: Software-Evolution (3LP)

SWE: Wissensmanagement in der Softwareentwicklung (2LP)

Veranstaltung	Semester	LP
Wiss. Arbeiten	1	2
Seminar – Lehrgebiet SWE	1-3	4
Seminar – Lehrgebiet DB	1-3	4
ISE Projekt	1-3	16
2 Vertiefende Module Lehrgebiet SWE	1-3	16
2 Vertiefende Vorlesungen Lehrgebiet DB	1-3	16
Weitere Module	1-3	6
Effiziente Algorithmen 1	1-3	8
Masterarbeit SWE oder DB	4	30
Anwendungsgebiet	1-3	18
LP Summe		120

Wurde das ISE-Projekt schon im Bachelor gemacht, so sollten im Master je 1 Praktikum in DB und SWE gemacht werden. Bei Belegung der Vertiefung im Bachelor müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

### ***Vertiefung Optimierung***

Diese Vertiefung hat als Inhalt Entwurf, Analyse und Implementierung von Algorithmen zur Lösung von kontinuierlichen und ganzzahligen Optimierungsproblemen. Sie kann im Bachelor- oder Master-Studium sowie durchgängig vom Bachelor- zum Master-Studium gewählt werden. Sie umfasst neben Vorlesungen, Seminaren und Praktika auch die Bachelor- bzw Masterarbeit. Empfohlene Voraussetzungen sind gute bis sehr gute Kenntnisse aus den Pflichtmodulen Mathematik.

Den Kern der Veranstaltungen dieser Vertiefung bilden die folgenden Module aus dem Gebiet Optimierung

Effiziente Algorithmen 1 (IEA1)

Effiziente Algorithmen 2 (IEA2)

Algorithmische Optimierung 1 (MH16)

Algorithmische Optimierung 2 (MH17)

sowie aus dem Gebiet Wissenschaftliches Rechnen  
Numerische Optimierung bei Differentialgleichungen (MH8)

Alle diese Module können dabei unabhängig voneinander absolviert werden.  
Wird die Vertiefung sowohl im Bachelor- als auch im Masterstudium gewählt, sollten alle 5 Module absolviert werden.

Weitere erforderliche Module sind  
mindestens ein Seminar (4 LP),  
mindestens ein Fortgeschrittenen-Praktikum (8 LP).

Empfohlen werden, je nach aktuellem Angebot, weitere Module wie z.B.  
Kompaktkurs Gemischt-ganzzahlige und kombinatorische Optimierung (MIP) und  
Modul Numerik (MD1) (besonders für Optimierung bei Differentialgleichungen).  
Möglich sind weiterhin

Lineare Optimierung (MD3)  
Nichtlineare Optimierung (MD4)  
Numerische Optimierung bei Differentialgleichungen (MH8)  
Numerische Methoden der Kontinuumsmechanik (MH9)  
Spezielle Themen der Numerik (MH10)  
Spezielle Themen der Optimierung (MH11)  
Mixed-Integer Programming and Combinatorial Optimization (IMIP)

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
Seminar Optimierung	1-3	4
Seminar (muss nicht Optimierung sein)		4
Fortgeschrittenen-Praktikum Optimierung	1-3	8
3 aus den oben genannten 5 Modulen	1-3	24
Weitere Module aus dem Gebiet Optimierung	1-3	8
Weitere Module aus mindestens 2 anderen Gebieten	1-3	22
Masterarbeit Optimierung	4	30
Anwendungsgebiet	1-3	18
LP Summe		120

Für weitere Module werden besonders die Gebiete Wissenschaftliches Rechnen, Bildverarbeitung und Datenbanken empfohlen. Bei Belegung der Vertiefung im Bachelor müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

### **Vertiefung Theoretische Informatik**

Die Vertiefung kann im Master gewählt werden. Sie umfasst Seminare, Wahlpflichtveranstaltungen und die Masterarbeit. Es sind 2 Seminare und 3 vertiefende Vorlesungen aus dem Gebiet der Theoretischen Informatik zu wählen und die Masterarbeit in diesem Gebiet anzufertigen. Weiter ist die Vorlesung Effiziente Algorithmen 1 verpflichtend. Weitere Lehrveranstaltungen in der Informatik sowie das Anwendungsgebiet können gemäß der Prüfungsordnung frei gewählt

werden.

Vertiefende Vorlesungen aus dem Bereich Theoretische Informatik (aus dem Modulhandbuch des Masters)

MH14 (8LP): Berechenbarkeit und Komplexität 1

MH15 (6LP): Berechenbarkeit und Komplexität 2

IFSA (8LP): Formale Sprachen und Automatentheorie

Auf Antrag kann eine dieser Vorlesungen durch eine vertiefende Vorlesung über effiziente Algorithmen ersetzt werden.

Ein Beratungsgespräch im 1. Semester mit einem der Dozenten der Theoretischen Informatik zur Ausgestaltung der Vertiefung wird empfohlen.

Veranstaltung	Semester	LP
Wiss. Arbeiten	1	2
2 Seminare – Lehrgebiet Theoretische Informatik	1-3	8
3 Vertiefende Vorlesungen - Lehrgebiet Theoretische Informatik	1-3	22
Effiziente Algorithmen 1	1-3	8
Weitere Module Informatik (Wahlpflichtmodule)	1-3	32
Anwendungsgebiet	1-3	18
Masterarbeit – Lehrgebiet Theoretische Informatik	4	30
LP Summe		120

### **Vertiefung Wissenschaftliches Rechnen**

Die Vertiefung kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst die Seminare, Praktika, Bachelor- und Masterarbeiten und die Wahlpflichtveranstaltungen.

Vertiefende Vorlesungen aus dem Gebiet Wissenschaftliches Rechnen (aus dem Modulhandbuch des Masters)

Numerik von Transportprozessen in porösen Medien (INTPM)
Objektorientiertes Programmieren im Wissenschaftlichen Rechnen (IOPWR)
Paralleles Höchstleistungsrechnen (IPHR)
Parallele Lösung großer Gleichungssysteme (IPLGG)
Simulationswerkzeuge (ISIMW)
Numerik (MD1)
Wissenschaftliches Rechnen (MD5)
Numerische Lineare Algebra (MH5)
Numerik partieller Differentialgleichungen (MH7)

Falls die Vertiefung noch nicht im Bachelor gewählt wurde, sollte der Modul MD1 gewählt werden

Veranstaltung	Semester	LP
Wiss. Arbeiten	1	2

Veranstaltung	Semester	LP
2 Seminare Wissenschaftliches Rechnen	1-3	8
Objektorientiertes Programmieren im Wissenschaftlichen Rechnen IOPWR	1-3	6
Numerik partieller Differentialgleichungen (MH7)	1-3	8
Numerische Optimierung bei Differentialgleichungen (MH8)	1-3	8
Paralleles Höchstleistungsrechnen (IPHR)	1-3	8
Algorithmische Optimierung I (MH16)	1-3	8
Eine weitere Veranstaltung aus: MH17, IPLGG, ISIMW, INTPM, IMIP	1-3	8
Weitere Module aus mindestens 1 anderem Gebiet (empfohlen: SWE )	1-3	16
Anwendungsgebiet	1-3	18
Masterarbeit Wissenschaftliches Rechnen		30
LP Summe		120

Bei Belegung der Vertiefung im Bachelor müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

## Module aus der Informatik

Modulbezeichnung:	<b>Algorithmische Geometrie (IAGM)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IAGM
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Heike Leite
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS, 1 SWS Übung
Arbeitsaufwand:	120 h 45 h Präsenzstudium 15 h Prüfungsvorbereitung 60 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)
Kreditpunkte:	4 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Algorithmen und Datenstrukturen
Lernergebnisse:	Die Studierenden kennen die grundlegenden Algorithmen und Datenstrukturen der geometrischen Datenverarbeitung. verstehen Grundkonzepte wie konvexe Hülle, Polygontriangulierung und effektive Punktsuche und sind in der Lage diese algorithmisch umzusetzen. beherrschen wesentliche Datenstrukturen zur effizienten Speicherung und Weiterverarbeitung der Daten. können die Komplexität der verschiedenen Algorithmen berechnen.
Inhalt:	Basiskonzepte Effiziente Punktsuche Voronoidiagramme Delaunaytriangulierung Allgemeine Suchstrukturen
Studien-/ Prüfungsleistung:	Bestehen einer schriftlichen oder mündlichen Abschlussprüfung
Medienformen:	Folien / Tafel / Skript / Lehrbuch
Literatur:	J. O'Rourke: Computational Geometry in C, Cambridge University Press, 1998. H. Edelsbrunner: Geometry and Topology of Mesh Generation, Cambridge University Press, 2001. Mark de Berg, Otfried Cheong, Marc van Kreveld,

Mark Overmars: Computational Geometry - Algorithms and Applications, 3rd edition, Springer, 2008.

Aktuelle Fachveröffentlichungen

Modulbezeichnung:	<b>Computergraphik 1 (ICG1)</b>
ggf. Modulniveau:	
ggf. Kürzel:	ICG1
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dr. Susanne Krömker
Sprache:	Englisch oder Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS, Übung 2 SWS
Arbeitsaufwand:	180 h; davon 60 h Präsenzstudium, 100 h Aufgabenbearbeitung, 20 h Prüfungsvorbereitung
Kreditpunkte:	6 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Einführung in die Praktische Informatik (IPR), Programmierkurs (IPK)
Lernergebnisse:	Die Studierenden sind in der Lage, einfache Graphikfenster mit Inhalt zu belegen, sind mit 2D-Rastergraphik vertraut und in der Lage, mit Objekten in Weltkoordinaten über Graphikbibliotheken (OpenGL) zu interagieren, können mit homogenen Koordinaten umgehen und Projektions- sowie Transformationsmatrizen bestimmen. Sie kennen die verschiedenen Bufferkonzepte zur schnellen Darstellung realistischer Effekte, sind mit Farbkonzepten und Darstellungsmodellen vertraut, können lokale Lichtmodelle aus optischen Prinzipien herleiten und wissen, wie man Texturen für unterschiedliche Effekte einsetzt. Sie kennen außerdem die physikalischen Grundlagen für globale Lichtmodelle.
Inhalt:	Grundlagen der Graphikprogrammierung Koordinatensysteme Projektionen, Transformationen Zeichenalgorithmen Bufferkonzepte (z-Buffer, Double-Buffer, Stereo-Buffer) Shading und Lichtmodelle Graphikbibliothek OpenGL Direct Rendering (lokale Verfahren) Texturen und Abbildungsverfahren Globale Verfahren Raytracing (Povray)

	Volume Rendering
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
Medienformen:	Folien / Skript / Lehrbuch
Literatur:	J. D. Foley, A. Van Dam, S. K. Feiner, J . F. Hughes und R. L. Phillips: Computer Graphics: Principles and Practice. Addison-Wesley, 1996 D. Shreiner, M. Woo, J. Neider und T. Davis: OpenGL programming guide: the official guide to learning OpenGL, version 1.4. OpenGL Architecture Review Board, Addison-Wesley, 2004

Modulbezeichnung:	<b>Computergraphik 2 (ICG2)</b>
ggf. Modulniveau:	
ggf. Kürzel:	ICG2
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 4. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dr. Susanne Krömker
Sprache:	Englisch oder Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS, Übung 2 SWS
Arbeitsaufwand:	180 h; davon 60 h Präsenzstudium, 100 h Aufgabenbearbeitung, 20 h Prüfungsvorbereitung
Kreditpunkte:	6 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Einführung in die Praktische Informatik (IPR), Programmierkurs (IPK), Computergraphik 1 (ICG1)
Lernergebnisse:	Die Studierenden sind mit weiterführenden Lichtmodellen vertraut, können vertex- und fragmentbasierte Shader programmieren, kennen die Vor- und Nachteile verschiedener globaler photorealistischer Verfahren, können nichtphotorealistische Verfahren sinnvoll einsetzen und mit Splines als Linien und Flächen umgehen.
Inhalt:	Shading und Bidirectional Reflectance Distribution Function (BRDF), C for graphics (Cg), Radiosity, Photon Mapping, Nichtphotorealismus (NPR), Splines
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
Medienformen:	Folien / Skript / Lehrbuch
Literatur:	J. D. Foley, A. Van Dam, S. K. Feiner, J . F. Hughes und R. L. Phillips: Computer Graphics: Principles and Practice. Addison-Wesley, 1996



	Randima Fernando, Mark J. Kilgard: The Cg Tutorial. Addison-Wesley, 2003
--	---

Modulbezeichnung:	<b>Compilerbau (ICOM)</b>
ggf. Modulniveau:	
ggf. Kürzel:	ICOM
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Gerhard Reinelt
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240 h 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen
Lernergebnisse:	Die Studierenden kennen den prinzipiellen Aufbau eines Compilers haben die erforderlichen Grundlagen zu formalen Sprachen gelernt sind mit den Problematiken von lexikalischer, syntaktischer und semantischer Analyse vertraut kennen die Techniken zur Compilererstellung sind über Optimierungsmöglichkeiten informiert sind in der Lage, vorhandene Werkzeuge zum Compilerbau zu benutzen können die erworbenen Kenntnisse auf andere Anwendungsbereiche, wie z.B. Aufbereitung strukturierter Daten, übertragen
Inhalt:	Überblick über grundlegende Techniken Grundlagen der formale Sprachen Lexikalische Analyse Top-Down-Syntaxanalyse Bottom-Up-Syntaxanalyse Syntaxgesteuerte Übersetzung Semantische Analyse

	Die C-Maschine 3-Adress-Code Optimierung Datenfluss-Analyse
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
Medienformen:	Folien / Tafel / Skript / Lehrbuch
Literatur:	z.B. Aho, A.V., Lam, M.S., Sethi, R., Ullman, J.D.: Compilers: Principles, Techniques and Tools, Pearson - Addison-Wesley, 2006

Modulbezeichnung	<b>Computerspiele (ICS)</b>
ggf. Modulniveau:	
ggf. Kürzel:	ICS
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Jürgen Hesser
Sprache:	Deutsch/Englisch (richtet sich nach den teilnehmenden Studenten)
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 3 SWS, Übung 3 SWS
Arbeitsaufwand:	240 h 75 h Präsenzstudium 15 h Prüfungsvorbereitung 150 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen
Lernergebnisse:	Die Studierenden lernen die Konzepte von der informatischen Sicht kennen, was Computerspiele ausmacht und welche Herausforderungen damit verbunden sind. Sie lernen, wie man gute und effiziente Architekturkonzepte dafür entwickelt sowie wie man typische Probleme aus Graphik, Kollisionserkennung, Animation/Physik, Pfadplanung/KI umsetzt. Zudem lernen sie in den Übungen, wie man konkrete Spiele entwickelt, so dass sie in der Lage sind, eigene Spieleengines zu realisieren.
Inhalt:	Überblick über die Einteilung von Computerspielen Architektur von Game Engines

	<p>Vorstellung von OGRE als einer open-source Game Engine</p> <p>Graphik und Computerspiele: ein Überblick</p> <p>Kollisionserkennungstechniken</p> <p>Animationstechniken und Physik bei Computerspielen mit Fokus auf der open source Bibliothek Bullet</p> <p>Pfadplanung und KI</p>
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung
Medienformen:	Folien / Lehrbuch
Literatur:	<p>Gregory et al: Game Engine Architecture</p> <p>Ericson: Real-Time Collision Detection</p> <p>Eberly: Game Physics</p> <p>Millington: Artificial Intelligence for Games</p>

Modulbezeichnung:	<b>Foundations of Cryptography and Network Security 1 (ICNS1)</b>
ggf. Modulniveau:	
ggf. Kürzel:	ICNS1
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 4. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dr. Sebastian Gajek, Jan Seedorf
Sprache:	Deutsch / Englisch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS mit Hausaufgaben
Arbeitsaufwand:	<p>90 h</p> <p>30 h Präsenzstudium</p> <p>30 h Aufgabenbearbeitung (evtl. in Gruppen)</p> <p>30 h Selbststudium und Prüfungsvorbereitung</p>
Kreditpunkte:	3 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Theoretische Informatik, Betriebssysteme und Netzwerke
Lernergebnisse:	Die Studierenden beherrschen die theoretischen Grundlagen von Kryptographie und haben ein grundlegendes Verständnis von kryptographischen Verfahren erlangt. Die wichtigsten Algorithmen und Protokolle zur Verschlüsselung und digitaler Signatur, die eine Vielzahl von praktischen IT-Sicherheitsapplikationen fundamentieren, sind ihnen im Detail

	<p>bekannt. Sie sind mit den mathematischen Modellen vertraut, kryptographische Protokolle zu analysieren und formal auf ihre Sicherheit zu beweisen.</p> <p>Die Studierenden sind mit den gängigen Bedrohungen in heutigen Computernetzen vertraut. Sie sind in der Lage, die damit verbundenden Sicherheitsrisiken einzuschätzen und bewerten zu können. Ferner haben die Studierenden grundlegende Kenntnisse über gängige Sicherheitstechniken und Verfahren zum Schutz von Computernetzen.</p> <p>Die Studierenden können sowohl aktuelle Sicherheitsprobleme in der Praxis identifizieren als auch effiziente Schutzmaßnahmen erarbeiten.</p>
Inhalt:	<p>Im ersten Teil der Veranstaltung werden theoretische Grundlagen der Kryptographie (z.B. one-way functions, permutation, pseudo-random generators, symmetrische Verschlüsselung, asymmetrische Verschlüsselung, digitale Signatur) und gängige kryptographische Verfahren und Protokolle (z.B. RSA, AES, SSL, IPsec, SSH) behandelt. Aufbauend auf diesen Grundlagen werden im zweiten Teil der Vorlesung praktische Anwendungen von kryptographischen Verfahren in Computernetzen sowie weitere Kernthemen im Bereich der Netzwerksicherheit behandelt. Themen sind unter anderem Zertifikate / Public Key Infrastructure (PKI), DNSsec, Kerberos, Intrusion Detection Systems, Firewalls und malizöse Software.</p>
Studien-/ Prüfungsleistung:	Erfolgreiche Abgabe von Hausaufgaben und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung
Medienformen:	Folien / Tafel / Skript / Lehrbuch
Literatur:	z.B. Dieter Gollmann: Computer Security, John Wiley & Sons; 2nd Edition (2005); William Stallings: Network Security Essentials, Pearson International, 2nd Edition

Modulbezeichnung:	<b>Foundations of Cryptography and Network Security 2 (ICNS2)</b>
ggf. Modulniveau:	Bachelor / Master
ggf. Kürzel:	ICNS2
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 4. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dr. Sebastian Gajek, Dr. Martin Johns
Sprache:	Englisch

Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS mit Hausaufgaben
Arbeitsaufwand:	30 h Präsenzstudium 30 h Aufgabenbearbeitung (evtl. in Gruppen) 30 h Selbststudium und Prüfungsvorbereitung
Kreditpunkte:	3 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Theoretische Informatik, Betriebssysteme und Netzwerke, Web Programmierung, ICNS1 - hilfreich, aber nicht notwendig
Lernergebnisse:	<p>Die Studierenden haben ein grundlegendes Verständnis von „interaktiven“ kryptographischen Verfahren („Protokollen“) und beherrschen die dafür notwendigen theoretischen Grundlagen der Kryptographie. Die wichtigsten Zwei-/Mehr-Parteien-Protokolle zur privaten Berechnung einer Funktion, die eine Vielzahl von aktuellen IT-Sicherheitsproblemen adressieren (z.B. wie Daten vertrauenswürdig in der Cloud bearbeitet werden) als auch Gegenstand der Forschung sind, sind ihnen im Detail bekannt. Sie können mit den erlernten mathematischen Definitionen und Modellen interaktive, kryptographische Protokolle nicht nur eigenständig entwerfen, sondern auch formal auf ihre Sicherheit untersuchen.</p> <p>Die Studierenden sind mit den wichtigsten Sicherheitsproblemen im Bereich der Web- und Softwaresicherheit vertraut. Um diesen zu begegnen, beherrschen die Studierenden grundlegende Prinzipien der Konzeption und Entwicklung von sicheren Programmen. Weiterhin haben die Studierenden grundlegendes Verständnis der verschiedenen anzuwendenden Angreifermodelle und Bedrohungsszenarios auf Anwendungsebene. Das Ziel der Vorlesung ist es das Können zu vermitteln, Systeme zu entwerfen, die, anhand des erworbenen Repertoires an kryptographischen Methoden, sicheren Designprinzipien und -methoden der sicheren Programmierung, dem jeweilig gegebenen Bedrohungsszenario gewappnet sind.</p>
Inhalt:	Im ersten Teil der Veranstaltung werden die theoretischen Grundlagen der Kryptographie, mit dem Fokus auf interaktive Mehrparteienprotokolle unter konkurrierender Exekution, vorgestellt. Unter anderem werden Interaktive Beweissysteme (Zero-Knowledge Proofs of Knowledge/Membership (ZKPoKs), Probabilistically Checkable Proofs (PCPs)), generische Compiler für Multi-Party Computation (MPC), und ggf.

	<p>auch Berechnungen auf verschlüsselten Daten (Fully-Homomorphic Encryption) gelehrt.</p> <p>Aufbauend darauf, stehen im zweiten Teil der Vorlesung Sicherheitsprobleme auf der Anwendungsebene im Mittelpunkt: Der Schwerpunkt liegt hierbei auf den Feldern der Software- und Web-Sicherheit. Anhand von Verwundbarkeitsklassen wie Cross-site Scripting, SQL Injection oder Buffer Overflows werden Angreifermodelle, Verwundbarkeitsszenarien und Methoden der sicheren Programmierung und des sicheren Designs von Software behandelt. Für jede der diskutierten Verwundbarkeitsklassen werden die verursachenden unsicheren Code-Konstrukte behandelt, die Methoden der eigentlichen Ausnutzung der resultierenden Sicherheitslücke demonstriert und gezeigt, wie das Sicherheitsproblem pro-aktiv, also innerhalb des Programcodes, wie auch reaktive, also im Betrieb des Systems, vermieden werden kann.</p> <p>Abschließend, wird anhand von Web getriebenen Security Protokollen, wie OAuth und BetterAuth, demonstriert, wie kryptographische Primitive sicher in nicht-trivialen Web Szenarien realisiert werden können.</p>
Studien-/ Prüfungsleistung:	Erfolgreiche Abgabe von Hausaufgaben und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung
Medienformen:	Folien / Tafel / Skript / Lehrbuch
Literatur:	Oded Goldreich: Foundations of Cryptography I+II, Michal Zalewski: The Tangled Web

Modulbezeichnung:	<b>Funddokumentation mittels optischer 3D-Scanner (I3DOK)</b>
ggf. Modulniveau:	
ggf. Kürzel:	I3DOK
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 4. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dr. Hubert Mara
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Übung 2 SWS
Arbeitsaufwand:	60 h 30 h Präsenzstudium 30 h Prüfungsvorbereitung

Kreditpunkte:	2 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Einführung in die Praktische Informatik (IPR) oder Computergraphik 1 (ICG1) oder Visualisierung im Bereich Cultural Heritage (IVCH) oder Vermessungskunde (UFG)
Lernergebnisse:	Die Studierenden sind mit unterschiedlichen 3D-Nahbereichsscantechniken vertraut, können 3D-Modelle aufbereiten und kennen die Herangehensweise mit 3D Bildverarbeitung zur Erkennung von Merkmalen (Schrift).
Inhalt:	3D-Meßtechnik basierend auf dem Prinzip des Strukturierten Licht, Aufnahme und Verarbeitung von hochauflösenden 3D-Modellen „3D-image-processing pipeline“
Studien-/ Prüfungsleistung:	Teilnahme an den einleitenden Vorlesungen und den praktischen Übungen. Bestehen einer praktischen Abschlussprüfung.
Medienformen:	Folien / Lehrbuch
Literatur:	Clive Orton: Mathematics in Archaeology. Cambridge, MA, Cambridge University Press, 1982 Katsushi Ikeuchi, Daisuke Miyazaki (editors): Digitally Archiving Cultural Objects. Springer, 2007 Christian Hörr. Algorithmen zur automatisierten Dokumenten- tation und Klassifikation archäologischer Gefäße. Dissertation, TU Chemnitz, 2011, Hubert Mara, Multi-Scale Integral Invariants for Robust Character Extraction from Irregular Polygon Mesh Data , Dissertation, Uni Heidelberg 2012

Modulbezeichnung:	<b>Data Mining - Algorithmen und parallele Verfahren (IDM)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IDM
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 5. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Artur Andrzejak
Sprache:	Deutsch
Zuordnung zum	

Curriculum:	
Lehrform/SWS:	Vorlesung 2 SWS, Übung 2 SWS mit Diskussion der Übungsaufgaben.
Arbeitsaufwand:	180 h
Kreditpunkte:	6 LP
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Vorkenntnisse:	Programmierkenntnisse (z.B. in Java) und elementare Wahrscheinlichkeitsrechnung; Vorlesung „Knowledge Discovery in Databases (IKDD)“ ist empfohlen aber nicht notwendig
Lernergebnisse:	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>verstehen die Nützlichkeit und Anwendungsgebiete einzelner Verfahren des Data Mining</li> <li>verstehen die Ansätze der parallelen Programmierung und der parallelen Datenanalyse, insbesondere Map-Reduce und GPU-Programmierung</li> <li>kennen die Methoden und Algorithmen der Vorverarbeitung von Daten, insbesondere parallele Attributselektion und Dimensionsreduktion (PCA)</li> <li>haben grundlegendes Verständnis der Klassifikation, Regression und Clustering sowie kennen ausgewählte parallele Implementierungen dieser Verfahren</li> <li>sind in der Lage, die Ergebnisse zu evaluieren und haben Verständnis der Problematik der Überanpassung (Overfitting)</li> <li>haben Fähigkeit der praktischen Verwendung von Data Mining Tools, Bibliotheken und Frameworks wie KNIME, WEKA, GNU Octave, NumPy/SciPy, Hadoop</li> </ul>
Inhalt:	<p>In diesem Modul werden die Verfahren des Data Mining mit Fokus auf Parallelisierung, Programmierertools und ihre sinnvolle Verwendung vorgestellt. Zu den Inhalten zählen:</p> <ul style="list-style-type: none"> <li>Grundbegriffe und Übersicht der Data Mining Verfahren</li> <li>Einführung in Tools, Bibliotheken und Frameworks</li> <li>Grundlagen der parallelen Programmierung (Map-Reduce, GPUs, Actors) für die Datenanalyse</li> <li>Verfahren zur Vorverarbeitung der Daten inklusive Attributselektion und Dimensionsreduktion</li> <li>Klassifikation, Regression, Clustering</li> <li>Bayes'sche Netze</li> <li>Evaluation der Ergebnisse, Overfitting und ihre</li> </ul>



	Vermeidung
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer Abschlussprüfung
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	z.B. Ethem Alpaydin, <b>Maschinelles Lernen</b> , Oldenbourg Verlag, 2008 Trevor Hastie, Robert Tibshirani, Jerome Friedman, <b>The Elements of Statistical Learning: Data Mining, Inference, and Prediction</b> , Springer, 2009 ( <a href="#">online</a> ) Stephen Marsland, Machine Learning: An Algorithmic Perspective, CRC Press Inc., 2009 Ron Bekkerman, Misha Bilenko, John Langford, <b>Scaling Up Machine Learning</b> , Cambridge University Press, 2012

Modulbezeichnung:	<b>Data Warehouses (IDW)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IDW
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 5. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Michael Gertz
Sprache:	Deutsch/Englisch nach Vereinbarung
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS, Übungen 1 SWS
Arbeitsaufwand:	120 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 75 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)
Kreditpunkte:	4 LP
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Vorkenntnisse:	Algorithmen und Datenstrukturen (IAD), Datenbanken 1 (DB1)
Lernergebnisse:	Die Studierenden sind in der Lage, zu einer gegebenen Anwendung eine Data Warehouse Pipeline zu modellieren kennen die Komponenten der Architektur von Data Warehouse Systemen wissen, welche Arten von Indexe und Speicherstrukturen typische Data Warehouse Anfragen unterstützen

	kennen Verfahren der Anfragebearbeitung und (manueller) Optimierungstechniken wissen, wie materialisierte Sichten einzusetzen sind und wie diese zu pflegen sind sind mit dem Aufsetzen eines realen Data Warehouses und den administrativen Tasks vertraut kennen die wichtigsten Performance Benchmarks für Data Warehouse Anwendungen
Inhalt:	Einführung & Grundbegriffe Data-Warehouse-Architekturen Modellierung von Data Warehouses Index- und Speicherungsstrukturen Anfragen an Data Warehouses Anfrageverarbeitung und -optimierung Materialisierte Sichten
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	z. B.: Wolfgang. Lehner. Datenbanktechnologie für Data-Warehouse-Systeme. dpunkt.verlag, Heidelberg, 2003. Andreas Bauer, Holger Günzel. Data Warehouse Systeme – Architektur, Entwicklung, Anwendung. 3. Auflage, dpunkt.verlag, Heidelberg, 2008. Gunter Saake, Andreas Heuer, Kai-Uwe Sattler. Datenbanken: Implementierungstechniken. 2. Auflage, mitp-Verlag, Bonn, 2005.

Modulbezeichnung:	<b>Effiziente Algorithmen 1 (IEA1)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IEA1
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Gerhard Reinelt
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übungen 2 SWS
Arbeitsaufwand:	240 h 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)

Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen
Lernergebnisse:	Die Studierenden verstehen die grundlegenden Konzepte der Graphentheorie können Fragestellungen als kombinatorische Optimierungsprobleme modellieren können die Komplexität von Optimierungsproblemen analysieren kennen die Methoden zum Beweis der Korrektheit kombinatorischer Algorithmen und der Analyse ihrer Laufzeit kennen die grundlegenden algorithmischen Ansätze sind mit den Fragen der effizienten Implementierung vertraut haben Einblick in die vielfältigen Anwendungsgebiete der kombinatorischen Optimierung
Inhalt:	Grundbegriffe der Graphentheorie Grundlegende Graphenalgorithmen Optimale Bäume und Branchings Kürzeste Wege Das Zuordnungsproblem Maximale Flüsse Minimale Schnitte in ungerichteten Graphen Flüsse mit minimalen Kosten Matchingprobleme
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
Medienformen:	Folien / Tafel / Skript / Lehrbuch
Literatur:	z.B. Korte, B., Vygen, J.: Kombinatorische Optimierung, Springer, 2007 Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., Schrijver, A.: Combinatorial Optimization, Wiley, 1997

Modulbezeichnung:	<b>Effiziente Algorithmen 2 (IEA2)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IEA2
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik

Dozent(in):	Prof. Dr. Gerhard Reinelt
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übungen 2 SWS
Arbeitsaufwand:	240 h 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen, Absolvierung des Moduls IEA1 ist nützlich, aber nicht Voraussetzung
Lernergebnisse:	Die Studierenden können polynomiale und NP-schwere Optimierungsprobleme klassifizieren kennen die gesamte Bandbreite von Lösungsansätzen für schwierige Probleme können geeignete Modellierungen für Anwendungsprobleme selbst erstellen sind in der Lage, geeignete Lösungsverfahren auszuwählen können selbst Lösungsverfahren konzipieren und implementieren
Inhalt:	NP-schwere Optimierungsprobleme approximative Algorithmen und Heuristiken (Bin-Packing, Scheduling, Knapsack, Traveling Salesman) Relaxierungen (lineare, kombinatorische, semidefinite, Lagrange-Relaxierungen) Verfahren zur Bestimmung optimaler Lösungen (dynamische Optimierung, Branch-and-Bound, Branch-and-Cut) lineare 0/1- Optimierung (Modellierung, Schnittebenen) polyedrische Kombinatorik, Spaltengenerierung und Dekomposition Fallstudien (Traveling-Salesman_Problem, Max-Cut-Problem.) Es wird somit das gesamte Spektrum von Lösungsverfahren für schwierige kombinatorische Probleme vermittelt.
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
Medienformen:	Folien / Tafel / Skript / Lehrbuch

Literatur:	z.B. Korte, B., Vygen, J.: Kombinatorische Optimierung, Springer, 2007 Nemhauser, G.L., Wolsey, L.A.: Integer Combinatorial Optimization, Wiley, 1988
------------	--

Modulbezeichnung:	<b>Software-Praktikum für Fortgeschrittene (IFM)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IFM
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin
Dozent(in):	Dozenten/Dozentinnen der Informatik
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Praktikum 6 SWS
Arbeitsaufwand:	240 h; davon mind. 25 h Präsenzstunden 10 h Vorbereitung Vortrag
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	
Lernergebnisse:	<p>Die Studierenden erlangen vertiefende Problemlösungskompetenz für komplexe Entwurfs- und Implementierungsaufgaben können Problemanalyse- und Beschreibungstechniken klar darstellen, differenzieren und anwenden vertiefen Programmierkenntnisse in der jeweiligen für das Projekt erforderlichen Programmiersprache sind in der Lage, das Projekt mit Hilfe einer Softwareentwicklungsumgebung durchzuführen</p> <p>Zusätzlich werden die projekttypischen Kompetenzen vertieft, insbesondere das Arbeiten im Team (von bis zu drei Studierenden): Durchführung und Evaluation von Projekten und ihrer Phasenstruktur Planung und Durchführung von Projekt- und Teamarbeit. Zu den zu trainierenden Softskills zählen somit insbesondere Teamfähigkeit, Verfeinerung von Präsentationstechniken, etwaige Erschließung wissenschaftlicher Literatur sowie</p>

	eigenverantwortliches Arbeiten.
Inhalt:	Domänenkenntnisse abhängig von den DozentInnen; allgemeine Lerninhalte sind: Vertiefung in die Projektarbeit Eigenständige Entwicklung von komplexer Software und deren Dokumentation
Studien-/ Prüfungsleistung:	Bewertung der dokumentierte Software, des Projektberichts und des Vortrag
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	

Modulbezeichnung:	<b>Formale Sprachen und Automatentheorie (IFSA)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IFSA
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Klaus Ambos-Spies, Priv.-Doz. Dr. Wolfgang Merkle
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übungen 2 SWS mit Hausaufgaben.
Arbeitsaufwand:	240 h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	ITH
Lernergebnisse:	Die Studierenden kennen die Stufen der Chomsky-Hierarchie und deren äquivalente Beschreibung durch Grammatiken und Automatenmodelle, kennen die Charakterisierungen der Klasse der regulären Sprachen durch rechts- bzw. linkslineare Grammatiken, durch deterministische und nichtdeterministische endliche Automaten, durch reguläre Ausdrücke und durch Rechtskongruenzen mit endlichem Index, sind in der Lage, das Pumping-Lemma für reguläre Sprachen zu beweisen und dieses anzuwenden, um zu zeigen, dass bestimmte Sprachen nicht regulär sind, kennen den Nachweis der Eindeutigkeit des Minimalautomaten und können diesen in Form eines

	<p>reduzierten Automaten oder eines Äquivalenzklassenautomaten konstruieren, können das Pumping-Lemma für kontextfreie Sprachen beweisen und dieses anwenden, um zu zeigen, dass bestimmte Sprachen nicht kontextfrei sind,</p> <p>können die Klasse der kontextfreien Sprachen durch kontextfreie Grammatiken und durch Kellerautomaten beschreiben und die Äquivalenz dieser Darstellungen beweisen,</p> <p>kennen den Satz von Chomsky und Schützenberger und seine anschauliche Bedeutung für die Struktur der Ableitungen in kontextfreien Sprachen, sind mit Normalformen, Abschlusseigenschaften, Entscheidbarkeits- und Komplexitätsaspekten insbesondere der regulären und kontextfreien Sprachen vertraut,</p> <p>können die Klasse der kontextsensitiven Sprachen durch kontextsensitive Grammatiken und Grammatiken vom Erweiterungstyp sowie durch linear beschränkte Automaten beschreiben und die Äquivalenz dieser Darstellungen beweisen.</p>
Inhalt:	Die Vorlesung gibt eine Einführung in die Theorie der formalen Sprachen und die dort verwendeten Begriffe und Methoden zur syntaktischen Sprachbeschreibung und -analyse unter besonderer Berücksichtigung algorithmischer Aspekte. Der Schwerpunkt der Vorlesung liegt auf den Sprachklassen der Chomsky-Hierarchie und deren Charakterisierung hinsichtlich der Beschreibbarkeit durch Grammatiken und der Erkennbarkeit durch Automaten.
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer mündlichen Abschlussprüfung.
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	

Modulbezeichnung:	<b>Flow and Transport in Terrestrial Systems (IFTTS)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IFTTS
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dr. Olaf Ippisch, Prof. Dr. Kurt Roth,

	Prof. Dr. Hans-Jörg Vogel
Sprache:	Englisch
Zuordnung zum Curriculum:	Wird unregelmäßig angeboten
Lehrform/ SWS:	Blockvorlesung mit Übung 5tägig (2 SWS)
Arbeitsaufwand:	90 h 40 h Präsenzstudium 50 h Selbststudium und Nachbereitung
Kreditpunkte:	3 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Grundkenntnisse über terrestrische Systeme (MVEEnv2 oder INTPM) oder die Numerik partieller Differentialgleichungen (MH7).
Lernergebnisse:	Die Studierenden haben die in natürlichen terrestrischen Systemen ablaufenden Transportprozesse gründlich verstanden. kennen die Grenzen der üblichen Modelle dieser Prozesse. können moderne numerische Simulationswerkzeuge effektiv einsetzen.
Inhalt:	Fundamentale physikalische Prozesse und aktuelle Konzepte zum Transport von Wasser und gelösten Stoffen im Untergrund Mehrskalige Architektur sedimentärer Formationen Makroskopische Phänomene und effektive Beschreibungen Numerische Simulation von Fluss und Transport in poröser Medien Wechselwirkung Boden-Atmosphäre.
Studien-/ Prüfungsleistung:	Mündliche oder schriftliche Abschlussprüfung (je nach Teilnehmerzahl)
Medienformen:	Folien / Tafel / Skript
Literatur:	

Modulbezeichnung:	<b>Grundlagen der wissenschaftlichen Visualisierung (IGVV)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IGVV
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Heike Leite



Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240 h 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen
Lernergebnisse:	Die Studierenden verstehen wie die menschliche visuelle Wahrnehmung abläuft und worauf man bei der visuellen Darstellung von Information aus wahrnehmungspsychologischer Sicht achten muss. kennen die Verarbeitungskette in der Visualisierung. kennen die grundlegenden Algorithmen zur Visualisierung von räumlichen und nicht-räumlichen Skalar- und Vektordaten. verstehen die Designentscheidungen bei der Implementierung von interaktiven System zur Datenexploration und können diese praktisch umsetzen. können selbständig Visualisierungsalgorithmen und GUI-basierte Anwendungsprogramme implementieren.
Inhalt:	Geschichte der Visualisierung Menschliche Wahrnehmung Verarbeitungskette in der Visualisierung Visualisierung von Skalar- und Vektordaten Visualisierung von Graphen Interaktion und Datenexploration
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung
Medienformen:	Folien / Tafel / Skript / Lehrbuch
Literatur:	Alexandru C. Telea: Data Visualization - Principles and Practice, A K Peters Ltd., 2007. Robert Spence: Information Visualization, Addison Wesley, 2000. Colin Ware: Information Visualization, Morgan Kaufmann, 2. Edition, 2004.

Modulbezeichnung:	<b>Inverse Problems (IIP)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IIP
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Jürgen Hesser
Sprache:	Englisch
Zuordnung zum Curriculum:	(scientific computing – wenn diese Vertiefung aktuell ist, sonst Wahlfach)
Lehrform/ SWS:	Vorlesung 2 SWS, Übung 2 SWS und Hausarbeiten
Arbeitsaufwand:	240 h 60 h Präsenzstudium 15 h Prüfungsvorbereitung 165 h Selbststudium und Aufgabenbearbeitung, Hausarbeiten
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen, Numerische Mathematik
Lernergebnisse:	Die Studierenden verstehen, was inverse Probleme sind und warum sie schwer zu lösen sind. Sie lernen die Prinzipien kennen, wie man sowohl deterministische Probleme, als auch stochastische Probleme löst und dabei auch die Regularisierungsparameter geeignet wählt. Schließlich erfahren sie die neuesten Entwicklungen im Bereich compressed sensing. Alle Prinzipien werden an zwei ausgewählten Gebieten, der Tomographie und des Deblurrings dargestellt. Sie erhalten damit die Kompetenz komplexe Probleme zu lösen, die mit klassischen Techniken nicht stabil lösbar sind und sind damit in der Lage auch komplexe experimentelle Messungen adäquat auswerten zu können
Inhalt:	Deterministische inverse Probleme Stochastische inverse Probleme Wahl der Regularisierungsparameter Compressed sensing Tomographie Deblurring

Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 60% der Punkte müssen erreicht werden) und Bestehen einer mündlichen Abschlussprüfung
Medienformen:	PDF-Folien, Lehrbuch, Matlab-Beispiele
Literatur:	M. Bertero, P. Boccacci: Introduction to Inverse Problems in Imaging, IoP, 2002 web-Page and book: <a href="http://www.slaney.org/pct/pct-toc.html">http://www.slaney.org/pct/pct-toc.html</a>

Modulbezeichnung:	<b>Knowledge Discovery in Databases (IKDD)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IKDD
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 5. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Michael Gertz
Sprache:	Deutsch/Englisch nach Vereinbarung
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übungen 2 SWS
Arbeitsaufwand:	240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Vorkenntnisse:	Algorithmen und Datenstrukturen (IAD), Datenbanken 1 (DB1)
Lernergebnisse:	Die Studierenden sind in der Lage, die Daten zu einem Data Mining Verfahren mit Hilfe geeigneter (statistischer) Methoden vorzuverarbeiten kennen die grundlegenden Verfahren der statistischen Datenanalyse sind in der Lage, grundlegende Techniken des Data Mining auf Daten aus verschiedenen Anwendungsbereichen anzuwenden sind in der Lage, Data Mining Techniken im Kontext von Datenbanken und des KDD Prozesses zu realisieren und anzuwenden kennen die Techniken und Prinzipien, die den Algorithmen zur Klassifikation und dem Clustering von Daten zugrunde liegen wissen, wie welche Clustering Algorithmen im Kontext

	<p>welcher Daten und Anwendungen zu verwenden sind kennen die Verfahren zum Finden häufiger Muster in Daten.  kennen die wichtigsten Techniken zur Erkennung von Ausreißern  sind vertraut mit den grundlegenden Algorithmen und Techniken zur Analyse von zeit- und raumbezogenen Daten</p>
Inhalt:	<p>Grundbegriffe: Statistik und Daten  Datenaufbereitung  Clustering  Frequent Pattern Mining  Klassifikationsverfahren  Mining von Graphen  Mining von räumlich und zeitlich veränderlichen Daten (z.B. Objekt-Trajektorien und Zeitreihen)</p>
Studien-/ Prüfungsleistung:	<p>Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung</p>
Medienformen:	<p>Folien/ Tafel/ E-Learning/ Lehrbuch</p>
Literatur:	<p>z. B.: Jiawei Han und Micheline Kamber: Data Mining. Concepts and Techniques, Morgan Kaufmann Series in Data Management Systems, 2006.  Martin Ester und Jörg Sander: Knowledge Discovery in Databases: Techniken und Anwendungen, Springer, 2000.</p>

Modulbezeichnung:	<b>Künstliche Intelligenz (IKI)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IKI
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Björn Ommer
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Wird unregelmäßig angeboten
Lehrform/ SWS:	Vorlesung 2 SWS, Übung 2 SWS
Arbeitsaufwand:	180 h 60h Präsenzstudium 15h Prüfungsvorbereitung 105h Selbststudium und Bearbeitung von theoretischen Aufgaben und Programmierübungen
Kreditpunkte:	6 LP

Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	
Lernergebnisse:	Die Studierenden haben gelernt Algorithmen zu entwickeln die Probleme durch Suchen lösen können logische Inferenz zur Lösungssuche einsetzen sind in der Lage Inferenz im Fall von unsicherem Wissen über die Welt durchzuführen können lernende Agenten entwickeln können die erworbenen Kenntnisse auf Anwendungsbereiche, wie z.B. Computersehen oder Textmining, übertragen
Inhalt:	Solving problems by searching Game playing Inference using logic Knowledge bases Planning and acting Reasoning under uncertainty Machine Learning Applications
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen und Bestehen einer Abschlussprüfung
Medienformen:	Folien / Tafel / Lehrbuch
Literatur:	Russell & Norvig: Artificial Intelligence – A Modern Approach

Modulbezeichnung:	<b>Konsistenzmodelle für parallele und verteilte Systeme (IKPV)</b>
ggf. Modulniveau:	
ggf. Kürzel	IKPV
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. A. Reuter
Sprache:	Deutsch / Englisch nach Vereinbarung
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS
Arbeitsaufwand:	60 h insgesamt, davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 15 h Selbststudium und Aufgabenbearbeitung
Kreditpunkte:	2 LP
Voraussetzungen nach Prüfungsordnung:	Keine

Empfohlene Vorkenntnisse:	Kenntnisse zu Algorithmen und Datenstrukturen, Datenbanksysteme
Lernergebnisse:	Die Studierenden bekommen ein Verständnis für die Probleme, die bei der Ausführung nebenläufiger Berechnungen auf gemeinsamen Ressourcen entstehen. Sie lernen die verschiedenen Ansätze zur Lösung dieser Probleme kennen und bekommen Kriterien dafür an die Hand, wie für einen gegebenen Anwendungsbereich die beste Balance zwischen Systemleistung und Konsistenz gefunden werden kann.
Inhalt:	Klassen verteilter und paralleler Algorithmen; Gleichzeitigkeit in verteilten Systemen; Arten der Wechselwirkung zwischen parallelen Aktivitäten; die Rolle gemeinsamer Ressourcen; die Rolle replizierter Ressourcen; relevante Fehlerklassen; Möglichkeiten zur Definition von Konsistenz (syntaktisch/semantisch); Arten konsistenzhaltender Ausführungen; Abwägung Konsistenz vs. Parallelität.
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an einer mündlichen Prüfung
Medienformen:	Folien / Tafel / E-Learning / Lehrbuch
Literatur:	z.B. Distributed Systems: Concepts and Design, by G. Coulouris, J. Dollimore and T. Kindberg (Addison Wesley Press) Transaction Processing-Concepts and Techniques, by Jim Gray and Andreas Reuter, Morgan Kaufmann Publishers Atomic Transactions, by Nancy Lynch et al, Elsevier Publishers

Modulbezeichnung:	<b>Logik in der Informatik (ILI)</b>
ggf. Modulniveau:	
ggf. Kürzel:	ILI
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Wolfgang Schönfeld
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS
Arbeitsaufwand:	90 h

	30 h Präsenzstudium 10 h Prüfungsvorbereitung 50 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)
Kreditpunkte:	3 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	keine
Lernergebnisse:	Die Studierenden haben Grundkenntnisse in mathematischer Logik, wissen wo diese in der Informatik eingesetzt werden, wissen wie Kalküle aufgebaut sind, können mit diesen Fragen logisch korrekt beantworten kennen die Grundeigenschaften von Kalkülen
Inhalt:	Syntax und Semantik der Aussagenlogik Syntax und Semantik der Prädikatenlogik Kalküle (Sequenzenkalkül, Tableaulkalkül) Beweissuchverfahren und ihre Implementierung
Studien-/ Prüfungsleistung:	Bestehen einer Abschlussprüfung
Medienformen:	elektronische Folien, Tafel, Bücher
Literatur:	U.Schöning, Logik für Informatiker , Spektrum Akademischer Verlag 2000 J. Kelly, Logik im Klartext, Pearson Studium 2003

Modulbezeichnung:	<b>Masterarbeit (IMa)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IMa
ggf. Lehrveranstaltungen:	
Studiensemester:	4. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dozenten/ Dozentinnen der Informatik
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Betreutes Selbststudium 2 SWS, Kolloquium 1 SWS
Arbeitsaufwand:	900 h, davon 810 h Bearbeitung eines individuellen Themas (Forschungs- und Entwicklungsarbeiten) und schriftliche Ausarbeitung 90 h Ausarbeitung Vortrag und Präsentation und Mitwirkung Kolloquium
Kreditpunkte:	30
Voraussetzungen nach Prüfungsordnung:	Keine

Empfohlene Vorkenntnisse:	Wahlpflichtvorlesungen und Module IS und IFP
Lernergebnisse:	Einsatz der erlernten Fachkenntnisse und Methoden zum selbstständigen Lösen einer komplexen Problemstellung aus der Informatik und ihren Anwendungen Fähigkeit, in großem Umfang selbstständig eine anspruchsvolle wissenschaftlichen Arbeit zu erstellen Fähigkeit, eine eigene, anspruchsvolle Arbeiten in einem wissenschaftlichen Vortrag darzustellen
Inhalt:	selbstständiges wissenschaftliches Bearbeiten einer beschränkten Aufgabenstellung aus der Informatik und ihren Anwendungen wissenschaftlicher Vortrag über die Problemstellung und die erarbeitete Lösung
Studien-/ Prüfungsleistung:	regelmäßige Treffen mit der/dem BetreuerIn, schriftliche Ausarbeitung und Präsentation
Medienformen:	
Literatur:	

Modulbezeichnung:	<b>Management und Analyse von Datenströmen (IMADS)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IMADS
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Michael Gertz
Sprache:	Deutsch/Englisch nach Vereinbarung
Zuordnung zum Curriculum:	
Lehrform / SWS:	Vorlesung 2 SWS, Übungen 1 SWS
Arbeitsaufwand:	120 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 75 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)
Kreditpunkte:	4 LP
Voraussetzung nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Algorithmen und Datenstrukturen (IAD), Datenbanken 1 (DB1)
Lernergebnisse:	Dieses Modul vertieft die Grundkenntnisse und -fähigkeiten



	<p>der Datenanalyse insbesondere im Umgang mit Datenströmen.</p> <p>Die Studierenden kennen die Herausforderungen der Datenstromverarbeitung und sind in der Lage, gängige Methoden anzuwenden, um Datenströme zu verwalten und auszuwerten.</p> <p>wissen, welche Mining Techniken für Datenströme im Kontext von verschiedenen Domänen anwendbar sind</p> <p>kennen die unter Inhalt angegebenen Methoden und Technologien</p>
Inhalt:	<p>Methoden des Managements und der Analyse von Datenströmen.</p> <p>Datenstrommodelle</p> <p>Datenstrommanagementsysteme</p> <p>Anfragesprachen und Anfrageverarbeitung</p> <p>Synopsenbildung und Approximation</p> <p>Data Mining Techniken wie beispielsweise Clustering, Frequent Pattern Mining und Erkennung von Ausreißern</p> <p><b>Mining und Analyse von Graph-Strömen</b></p>
Studien- / Prüfungsleistung:	Erfolgreicher Abschluss eines Projekts und Bestehen einer Abschlussprüfung
Medienformen:	Folien / Tafel / E-Learning / Lehrbuch
Literatur:	Wissenschaftliche Veröffentlichungen. Details werden in der Vorlesung und auf der Moodle-Webseite bekannt gegeben.

Modulbezeichnung:	<b>Mixed-Integer Programming and Combinatorial Optimization (IMIP)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IMIP
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Gerhard Reinelt
Sprache:	Englisch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	1-wöchiger Kompaktkurs
Arbeitsaufwand:	60h

	20 h Präsenzstudium 10 h Prüfungsvorbereitung 30 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen) am Rechner
Kreditpunkte:	3 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Grundkenntnisse Mathematik
Lernergebnisse:	Die Studierenden kennen die Modellierungsmöglichkeiten der gemischt-ganzzahligen Optimierung sind mit einer kommerziellen MIP-Software vertraut haben einen guten Überblick über die Anwendungsmöglichkeiten sind mit den polyedertheoretischen Grundlagen der Optimierung vertraut kennen die wichtigsten Optimierungstechniken haben die erforderlichen Kenntnisse zur Beurteilung von Möglichkeiten und Grenzen der praktischen Verwendbarkeit der erlernten Techniken
Inhalt:	Linear programming (polyhedral theory, duality, simplex algorithms, postoptimal analysis) Mixed-integer modelling Combinatorial optimization problems Computation of optima (cutting plane, branch-and-bound, Branch-and-cut algorithms, dynamic programming) Polyhedral combinatorics Combinatorial polytopes Relaxations (LP, semidefinite, Lagrange) Implementation of branch-and-cut algorithms Valid inequalities Presolve Techniques for large problems (column generation, decomposition) Applications
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
Medienformen:	Folien / Tafel / Lehrbuch
Literatur:	z.B. Kallrath, J. and Wilson, J.M.: Business Optimisation using Mathematical Programming, Macmillan Press, 1997 Williams, H.P.: Model Building, Wiley, 1999

Modulbezeichnung:	<b>Neuroscientific Computing (INSC)</b>
-------------------	---

ggf. Modulniveau:	Bachelor / Master
ggf. Kürzel:	INSC
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 4. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dr. Stefan Lang
Sprache:	Deutsch / Englisch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS, Übung 2 SWS
Arbeitsaufwand:	180 h 60 h Präsenzstudium 25 h Prüfungsvorbereitung 95 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)
Kreditpunkte:	6 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Grundlagen in Numerik wie z.B. aus MA7
Lernergebnisse:	Die Studierenden kennen den makro-/mikroskopischen Aufbau eines Gehirns und verstehen die Prinzipien neuronaler Signalverarbeitungsprozesse. sind in der Lage die Eigenschaften individueller Neuronenmodelle mit Methoden aus dem Bereich der dynamischen Systeme zu analysieren. können mathematische Modelle von Einzelneuronen selbst entwerfen und numerische Verfahren zur Lösung bestimmen. haben die Fähigkeit die erforderlichen numerischen Methoden in einer Programmiersprache (C/C++ ) eigenständig umzusetzen. haben die Kenntnis die Simulationsergebnisse mathematisch-informatisch zu analysieren und zu interpretieren.
Inhalt:	Grundkenntnisse in Neuroanatomie und -physiologie Bioelektrizität: Biophysikalische Grundlagen von Ionenströmen, Elektrophysiologie, Nernstpotential Biomembranmodelle Modellierung von Signalverarbeitungsprozessen im extrazellulären Raum Integrate-and-Fire Modelle / Neuronenmodelle mit einem Kompartiment Numerische Lösungsmethoden für Einkompartimentmodelle

	<p>Dynamische Systeme: Begriffsbildungen, Phasenraumanalyse, Stabilität und Hysterese, Bifurkationen, Spiking, Bursting</p> <p>Numerische Lösungsverfahren für steife Systeme</p> <p>Modellierung und Simulation der passiven Kabelgleichung</p> <p>Diskretisierung der Kabelgleichung</p> <p>Das Hodgkin-Huxley Modell (HH) der aktiven Signalverarbeitung</p> <p>Methoden zur nichtlineare Kopplung von HH und Kabelgleichung</p> <p>Simulation stochastischer Prozesse</p>
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung
Medienformen:	Folien / Tafel / Skript / Lehrbuch
Literatur:	<i>C. Koch: Biophysics of Computation: Information Processing in Single Neurons.</i> Oxford Univ. Press, 1999, <a href="https://doi.org/10.1017/CBO9780511527187">ISBN 0-19-518199-9</a>

Modulbezeichnung:	<b>Numerik von Transportprozessen in porösen Medien (INTPM)</b>
ggf. Modulniveau:	
ggf. Kürzel:	INTPM
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dr. Olaf Ippisch
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS, Übung 2 SWS am Rechner.
Arbeitsaufwand:	180 h
Kreditpunkte:	6 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	
Lernergebnisse:	<p>Die Studierenden sind in der Lage</p> <p>Strömungsprozesse in porösen Medien zu modellieren</p> <p>Transport gelöster Stoffe zu modellieren</p> <p>Grenzen und Voraussetzungen dieser Modelle zu verstehen</p> <p>die wichtigsten numerischen Methoden zur Simulation dieser Modelle anzuwenden</p>

	die vermittelten Modelle auf konkrete Probleme der Bodensanierung anzuwenden
Inhalt:	Grundlagen partieller Differentialgleichungen Grundlagen der Strömungsmechanik Grundwasserströmung und Darcy-Gesetz Heterogenitäten und deren Charakterisierung, präferentielle Fließpfade Transport gelöster Stoffe hydrodynamische Dispersion Finite Differenzen, Finite Volumen und Finite Elemente Verfahren iterative Lösung linearer Gleichungssysteme.
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung (diese kann bei geringer Teilnehmerzahl durch eine mündliche Prüfung ersetzt werden).
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	

Modulbezeichnung:	<b>Objekterkennung und automatisches Bildverstehen (IOAB)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IOAB
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Professoren des HCI
Dozent(in):	Professoren des HCI
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS , Übung 2 SWS
Arbeitsaufwand:	240 h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	
Lernergebnisse:	Die Studierenden sind auf dem neuesten Stand der Forschung im Bereich von Mid- und High-Level Computer Vision kennen die relevantesten Ansätze zur Merkmalsextraktion sind vertraut mit den wesentlichen Methoden zur robusten Objektrepräsentation haben essentielle Algorithmen aus dem Bereich der

	<p>Mustererkennung und des maschinellen Lernens verstanden und können diese auf neue Probleme übertragen und anwenden sind in der Lage aktuelle Forschungsarbeiten aus dem Themenbereich der Vorlesung zu verstehen und in den Kontext der klassischen Arbeiten einzuordnen können Objekterkennungsprobleme in neuen Anwendungen analysieren und die jeweils geeignete Algorithmik dazu abrufen um darauf aufbauend neue Lösungsansätze zu entwickeln sind fähig Objekterkennungsverfahren differenziert zu bewerten und die Validität einer experimentellen Evaluation zu überprüfen</p>
Inhalt:	<p>Methoden und Modelle aus dem Bereich von Mid- und High-Level Computer Vision. Insbesondere: Objektdetektion und –klassifikation lokale und globale Merkmalsextraktion Modell-basierte Ansätze Ansichten-basierte Methoden generative/diskriminative Verfahren Registrierung Formanalyse Voting Methoden Hashing Verfahren Hierarchische Objektrepräsentationen Erkennung von menschlichen Aktionen Lernverfahren für Objektrepräsentationen</p>
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen und Bestehen einer Abschlussprüfung.
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	

Modulbezeichnung:	<b>Objektorientiertes Programmieren im Wissenschaftlichen Rechnen (IOPWR)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IOPWR
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Peter Bastian
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 2 SWS, Übung 2 SWS am Rechner.

Arbeitsaufwand:	180 h
Kreditpunkte:	6 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Kenntnisse einer objektorientierten Programmiersprache
Lernergebnisse:	Die Studierenden beherrschen die Programmiersprache C++ sind in der Lage Performanz von unterschiedlichen Lösungen zu beurteilen beherrschen Template Programmiertechniken können die Standard Template Library einsetzen sind in der Lage die gelehrtten Konzepte an ausgewählten Problemen des Wissenschaftlichen Rechnens praktisch umzusetzen
Inhalt:	Dieses Modul vertieft die in der Grundvorlesung "Einführung in die Praktische Informatik" vermittelten Kenntnisse in objektorientierter Programmierung mit spezieller Ausrichtung auf das Wissenschaftliche Rechnen: Klassenkonzept Dynamische Speicherverwaltung Ausnahmebehandlung Ressourcenallokierung und Initialisierung Benutzung von const Template Metaprogrammierung, statischer vs. dynamischer Polymorphismus Traits Policies Standard Template Library
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (Erreichen von 50% der Punkte) und Bestehen einer Abschlussprüfung.
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	

Modulbezeichnung:	<b>Paralleles Höchstleistungsrechnen (IPHR)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IPHR
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Peter Bastian/ Dr. Stefan Lang
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS am Rechner mit

	Diskussion der Übungsaufgaben.
Arbeitsaufwand:	240 h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Kenntnisse einer Programmiersprache (Programmieren und Softwaretechnik) und grundlegender Algorithmen (Algorithmen und Datenstrukturen (IAD)).
Lernergebnisse:	Die Studierenden kennen die grundlegenden Architekturen paralleler Höchstleistungsrechner kennen die grundlegenden Synchronisationsmechanismen in parallelen Systemen inklusive Performanzaspekten beherrschen die wichtigsten Programmierparadigmen für parallele Systeme sind in der Lage grundlegende Synchronisationsaufgaben zu lösen beherrschen Algorithmen der parallelen linearen Algebra können die Qualität paralleler Algorithmen und deren praktische Implementierung bewerten
Inhalt:	Rechner mit globalem Adressraum Cache-Kohärenz Rechner mit lokalem Adressraum und Nachrichtenaustausch kritischer Abschnitt, Bedingungsynchronisation Semaphore Posix Threads Programmierung von Grafikkarten Nachrichtenaustausch MPI Client-Server Architekturen, entfernter Prozeduraufruf Bewertung paralleler Algorithmen Lastverteilung Algorithmen für vollbesetzte Matrizen Lösung dünnbesetzter Gleichungssysteme Partikelmethode Paralleles Sortieren
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (Erreichen von 50%) und Bestehen einer Abschlussprüfung.
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	

Modulbezeichnung:	<b>Parallele Lösung großer Gleichungssysteme</b>
-------------------	--



	<b>(IPLGG)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IPLGG
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Peter Bastian
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übung 2 SWS am Rechner mit Diskussion der Übungsaufgaben.
Arbeitsaufwand:	240 h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Kenntnisse einer objektorientierten Programmiersprache, MA7, Kenntnisse wie sie das Modul IPHR oder ICC1 vermittelt, Vorkenntnisse in Numerik, insbesondere Differentialgleichungen.
Lernergebnisse:	Die Studierenden beherrschen die Diskretisierung elliptischer Differentialgleichungen mittels Finiter Elemente verstehen das abstrakte Konzept der Teilraumkorrekturverfahren können dies im Rahmen von überlappenden und nichtüberlappenden Gebietszerlegungsverfahren sowie Mehrgitterverfahren umsetzen beherrschen die Konvergenztheorie zu diesen Verfahren können ausgewählte Verfahren praktisch auf Parallelrechnern umsetzen und der Leistungsfähigkeit bewerten
Inhalt:	Grundlagen der Diskretisierung elliptischer partieller Differentialgleichungen, Teilraumkorrekturverfahren überlappende und nichtüberlappende Gebietszerlegungsverfahren mit Konvergenztheorie geometrische Mehrgitterverfahren mit Konvergenztheorie algebraische Mehrgitterverfahren
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (Erreichen von 50% der Punkte) und Bestehen einer Abschlussprüfung.
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	

Modulbezeichnung:	<b>IT-Projektmanagement (IPM)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IPM
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dr. Privat-Doz. Andrea Herrmann
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung+Übung 2 SWS
Arbeitsaufwand:	60 h insgesamt, davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 15 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)
Kreditpunkte:	2 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Vorlesung und Übung „Einführung in Software Engineering“ (ISW) oder vergleichbare Vorkenntnisse
Lernergebnisse:	Die Teilnehmer/innen können ein Projekt planen und überwachen, verstehen wie Projekte in Organisationen eingebettet sind und haben Grundkenntnisse in vertraglichen Themen.
Inhalt:	Projektplanung, Projektorganisation Kostenschätzung Angebot/ Vertrag, Verhandeln Vorgehensmodelle Risikomanagement Controlling IT-Vertragsrecht Änderungsmanagement Zeitmanagement Projektabschluss Verteilte Softwareentwicklung
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Hausaufgaben (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer Abschlussprüfung.
Medienformen:	Folien & Tafel
Literatur:	PMI (Project Management Institute): A Guide to the Project Management Body of Knowledge (PM BOK® Guide), 4. Ausgabe 2008

Modulbezeichnung:	<b>Randomisierte Algorithmen (IRA)</b>
Modulniveau:	Bachelor / Master
Kürzel:	IRA
Lehrveranstaltungen:	-
Studiensemester:	Ab 4. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekanin/ Studiendekan der Informatik
Dozent:	PD Dr. Wolfgang Merkle
Sprache:	Deutsch (nach Absprache alternativ Englisch)
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 3 SWS, Übung 1 SWS
Arbeitsaufwand:	180 h 60 h Präsenzstudium 40 h Prüfungsvorbereitung 80 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)
Kreditpunkte:	6 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Elementare Grundkenntnisse in Algorithmen wie sie z.B. im Modul IAD vermittelt werden.
Lernergebnisse:	Auf der Grundlage der behandelten Anwendungsbeispiele aus verschiedenen Teilgebieten der Informatik können die Studierenden die probabilistische Betrachtungs- und Vorgehensweise anwenden - bei der Konstruktion und Analyse von probabilistischen und deterministischen Algorithmen, - auf kombinatorische Fragestellungen, - um spieltheoretische Situationen zu analysieren, - auf kryptographische Fragestellungen.
Inhalt:	Elementare Wahrscheinlichkeitsrechnung Das Tenure-Spiel Derandomisierungstechniken Die probabilistische Methode Byzantinische Übereinkunft Stabile Heiraten und der Gale-Shapley-Algorithmus Das Minimax-Prinzip von Yao Komplexitätsanalyse des randomisierten Sortierens Randomisierte Fehlersuche und -korrektur Das Local-Lemma von Lovasz PAC-Lernen und VC-Dimension Wahrscheinlichkeitsverstärkung und Fehlerschranken Lokale Suche für k-SAT Kryptographische Protokolle

Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen und Bestehen einer mündlichen oder schriftlichen Abschlussprüfung
Medienformen:	Folien / Tafel / Lehrbuch
Literatur:	R. Motwani und P. Raghavan, Randomized Algorithms, Cambridge University Press 1995. M. Mitzenmacher und E. Upfal, Probability and Computing, Cambridge University Press, 1995. N. Alon und J. H. Spencer, The Probabilistic Method, John Wiley and Sons, 2008.

Modulbezeichnung:	<b>Räumliche Datenbanken (IRDB)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IRDB
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 5. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Michael Gertz
Sprache:	Deutsch/Englisch nach Vereinbarung
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übungen 2 SWS
Arbeitsaufwand:	240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Algorithmen und Datenstrukturen (IAD), Datenbanken 1 (IDB1)
Lernergebnisse:	Die Studierenden kennen die Prinzipien von und Anforderungen an räumliche Daten und die Verwaltung solcher Daten in verschiedenen Anwendungsbereichen (z.B. in der Geographie, Kosmologie und Biologie) kennen die Konzepte und Anwendungen Geographischer Informationssysteme (GIS) sind in der Lage, Konzepte und Modelle für räumliche Daten und der Datenmodellierung anzuwenden sind vertraut mit der Unterstützung zur Verwaltung von und Anfrage an räumliche Datenbanken (z.B. PostGIS) kennen grundlegende Methoden der algorithmischen Geometrie und sind in der Lage, entsprechende Algorithmen und Techniken anzuwenden

	<p>kennen die wichtigsten Vertreter von Indexstrukturen zu räumlichen Daten (Gridfile, kd-Tree, Quadtree, R-Tree)</p> <p>wissen, wie in existierenden räumlichen Datenbanksystemen (z.B. PostGIS) Datenbankschemata und Anwendungen erstellt werden</p>
Inhalt:	<p>Prinzipien und Anforderungen an die Verwaltung räumlicher Daten</p> <p>Anwendungsbereiche zur Verwaltung und Analyse räumlicher Daten</p> <p>Unterstützung zur Verwaltung räumlicher Daten in kommerziellen und Open-Source DBMS</p> <p>Konzepte und Modelle zur Repräsentation räumlicher Daten (2D, 3D); Tessellierung und Vektormodell; Gruppen von räumlichen Objekten</p> <p>Abstrakte Datentypen für räumliche Daten</p> <p>Grundlegende Techniken der algorithmischen Geometrie (z.B. Konvexe Hülle, Sweep-Line Methoden, Polygon-Partitionierung, Schnitte von Polygonen)</p> <p>Zugriffsstrukturen für räumliche Daten, insbesondere Grid-Files, kd-Tree, Quadtrees, R-Tree</p> <p>Algorithmen und Kostenmodelle für Zugriffsstrukturen zu räumlichen Daten</p> <p>Konzepte der Anfrageverarbeitung und -optimierung für räumliche Datenbanken, insb. Spatial Join</p> <p>Temporale Datenbanken und Indexstrukturen</p> <p>Moving Objects: Anwendungen, Anfragen, Indexstrukturen (B<sup>x</sup>-Tree und TPR-Tree)</p> <p>Einführung in das Mining räumlicher Daten (Clustering, Entdecken von Ausreißern)</p> <p>Überblick über Unterstützung zur Verwaltung räumlicher Daten in kommerziellen und Open-Source DBMS (PostgreSQL, MySQL, Oracle Spatial, GRASS)</p>
Studien-/ Prüfungsleistung:	<p>Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung</p>
Medienformen:	<p>Folien/ Tafel/ E-Learning/ Lehrbuch</p>
Literatur:	<p>z. B.: Spatial Databases – With Applications to GIS. Philippe Rigaux, Michel Scholl, Agnes Voisard. Morgan Kaufmann, 2001.</p> <p>Computational Geometry: Algorithms and Applications Mark de Berg, Otfried Cheong, Marc van Kreveld, und Mark Overmars, Springer, Berlin, 2008.</p> <p>Forschungsartikel aus Tagungsbänden und Journals</p>

Modulbezeichnung:	<b>Seminar (IS)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IS
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor und 1.Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dozenten/ Dozentinnen der Informatik
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Seminar 2+2 SWS
Arbeitsaufwand:	120 h, davon
Kreditpunkte:	4 LP
Voraussetzungen nach Prüfungsordnung:	IPS
Empfohlene Vorkenntnisse:	Kenntnisse im Themengebiet des Seminars
Lernergebnisse:	Kenntnis von Techniken des wissenschaftlichen Schreibens (insbesondere auch Literaturrecherche) Fähigkeit, komplexe wissenschaftliche Literatur zu erschließen Erweiterte Fähigkeit, komplexe wissenschaftliche Literatur in einem Vortrag zu präsentieren Erweiterte Fähigkeit, zu Vorträgen zu diskutieren und Feedback zu geben Fähigkeit, ein kurze wissenschaftliche Ausarbeitung zu einem komplexen Thema zu erstellen
Inhalt:	Einführung in und Einübung von Techniken des wissenschaftlichen Schreibens Vertiefte Einübung der Erschließung und Präsentation wissenschaftlicher Literatur Fortgeschritteneres Informatikthema
Studien-/ Prüfungsleistung:	Regelmäßige Teilnahme an den Seminarterminen und Mitwirkung in den Diskussionen Ausarbeitung und Halten eines Vortrages von etwa 60 Minuten Dauer (inklusive Diskussion). schriftliche Ausarbeitung von ca. 10 Seiten
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	

Modulbezeichnung:	<b>Simulationswerkzeuge (ISIMW)</b>
ggf. Modulniveau:	
ggf. Kürzel:	ISIMW
ggf.	

Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Prof. Dr. Peter Bastian
Dozent(in):	Dr. Stefan Lang
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 2 SWS, Übung 2 SWS
Arbeitsaufwand:	180 h
Kreditpunkte:	6 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Grundlagen der Numerik
Lernergebnisse:	<p>Die Studierenden beherrschen grundlegende Verfahren zur Repräsentation komplexer Geometrien (CAD-Modelle) verstehen die Methoden zur Vernetzung komplexer Geometrien beherrschen standardisierte Datenformate zur Speicherung großer, wissenschaftlicher Datenmengen haben grundlegende Verfahren zur Visualisierung von Ergebnissen von Simulationsprogrammen kennengelernt können diese Verfahren im Kontext realistischer Anwendungsprobleme konkret einsetzen</p>
Inhalt:	<p>Einführung in die grundlegenden Methoden, Techniken und Softwarerealisierungen bei Entwurf und Repräsentation komplexer Geometrien und deren Vernetzung. Neben der reinen Simulation eines Vorganges, d.h. der numerischen Lösung des mathematischen Problems nehmen die Vor- und Nachverarbeitungsschritte einen zunehmend größeren Stellenwert bei der Durchführung von Simulationsstudien ein. Zur Vorverarbeitung zählen z.B. die Erstellung und Repräsentation komplexer, dreidimensionaler Geometrien mit Splines und Nurbs Zuordnung von Koeffizienten, z.B. Materialeigenschaften, und Randbedingungen Gittergenerierung mit Delauney und Advancing- Front-Methoden Speicherung größter Datenmengen, z.B. HDF5 Datenaustausch mit Visualisierungssystemen und anderen Programmen Konkrete Benutzung von Simulationswerkzeugen an konkreten Beispielen, z.B. Salome, OpenCascade, Gmsh, Paraview/VTK</p>
Studien-/	Erfolgreiche Teilnahme an den Übungen (Erreichen

Prüfungsleistung:	von 50% der Punkte aus den Übungsaufgaben).
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	

Modulbezeichnung:	<b>Software Evolution (ISWEvol)</b>
ggf. Modulniveau:	
ggf. Kürzel:	ISWEvol
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 4. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Prof.Dr. Barbara Paech
Dozent(in):	Dr. Eckhart von Hahn
Sprache:	Deutsch
Zuordnung zum Curriculum:	Angewandte Informatik sowie HörerInnen anderer Fachrichtungen
Lehrform/ SWS:	Vorlesung 2 SWS
Arbeitsaufwand:	90h, davon 30 h Vorlesung 35 Aufgabenbearbeitung und Aufarbeitung/Selbststudium 25 h Prüfungsvorbereitung (evtl. in Gruppen)
Kreditpunkte:	3 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Kenntnisse und Fähigkeiten wie sie in Modul ISW vermittelt werden.
Lernergebnisse:	<p>Die Studierenden können nach der Vorlesung ...</p> <ul style="list-style-type: none"> <li>... ein Software-Reengineering-Projekt fachlich planen und beurteilen,</li> <li>... bei der Ersterstellung von Software die Evolutionsfähigkeit konzeptuell sicherstellen,</li> <li>... ein Wartungskonzept für eine erstellte Software aufbauen.</li> </ul> <p>Sie kennen</p> <ul style="list-style-type: none"> <li>... die Unterschiede und Herausforderungen der Software-Weiterentwicklung versus der Softwareneuentwicklung – und worauf die/der InformatikerIn hierbei achten muss, sowohl aus Sicht eines Softwareherstellers als auch aus der Sicht der NutzerInnen von Software,</li> <li>... die klassischen Techniken der Softwaresanierung,</li> <li>... die Typologie der Softwarewartung und das Management der Fehlerbehebung,</li> <li>... die Relevanz der Thematik in der Praxis der</li> </ul>



	industriellen Softwareerstellung.
Inhalt:	<p>Dieses Modul vermittelt aus konzeptioneller Sicht die Grundlagen für ein erfolgreiches Lebenszyklusmanagement von Software nach ihrer Ersterstellung. Die Vorlesungsinhalte wurden aufgrund der 15-jährigen Erfahrung des Dozenten in der Praxis zusammengestellt, auf Basis aktueller Forschung und Lehre.</p> <p>Einleitung          Begriffsklärung, Grundlagen          Softwareevolution          Softwarewartung, Softwareerhaltung          Software-Reengineering          Evolution und Weiterentwicklung          Management der Softwareevolution          Zusammenfassung</p>
Studien-/Prüfungsleistung:	Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Anzahl der Teilnehmer)
Medienformen:	Folien/ Tafel/ E-Learning
Literatur:	<p>z.B.</p> <p>Arnold, R. (Hrsg.): Software Reengineering. IEEE Computer Society Press, Los Alamitos 1993</p> <p>Fowler, M.: Refactoring – Improving the Design of Existing Code. Addison-Wesley, Reading, Massachusetts, 1999</p> <p>von Hahn, E.: Werterhaltung von Software. DUV, Wiesbaden 2005</p> <p>Müller, B.: Reengineering. Eine Einführung. Teubner, Stuttgart 1997</p> <p>Sneed, H.M.; Hasitschka, M.; Teichmann, M.-T.: Software-Produktmanagement. Wartung und Weiterentwicklung bestehender Anwendungssysteme. dpunkt, Heidelberg 2005</p> <p>Smith, D.D.: Designing Maintainable Software. Springer, Heidelberg 1999</p>

Modulbezeichnung:	<b>Software Ökonomie (ISWÖk)</b>
ggf. Modulniveau:	
ggf. Kürzel:	ISWÖk
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 4. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Prof.Dr. Barbara Paech
Dozent(in):	Dr. Eckhart von Hahn

Sprache:	Deutsch
Zuordnung zum Curriculum:	Angewandte Informatik sowie HörerInnen anderer Fachrichtungen
Lehrform/ SWS:	Vorlesung 2 SWS
Arbeitsaufwand:	90h, davon 30 h Vorlesung 35 Aufgabenbearbeitung und Aufarbeitung/Selbststudium 25 h Prüfungsvorbereitung (evtl. in Gruppen)
Kreditpunkte:	3 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Kenntnisse und Fähigkeiten wie sie in Modul ISW vermittelt werden.
Lernergebnisse:	<p>Die Studierenden können nach der Vorlesung ...</p> <ul style="list-style-type: none"> <li>... grob den Preis und die Lizenzierung einer erstellten Software ermitteln,</li> <li>... die Vermarktung von Software planen und anstoßen,</li> <li>... grob die Bilanz sowie die Gewinn- und Verlustrechnung eines Softwareherstellers verstehen,</li> <li>... den Wert einer Software mit seinen verschiedenen Komponenten beurteilen, aus Sicht des Herstellers sowie aus Sicht der Nutzer,</li> <li>... Preisverhandlungen zu Softwareprojekten planen.</li> </ul> <p>Sie kennen</p> <ul style="list-style-type: none"> <li>... die Grundzüge der Kosten- und Leistungsrechnung (soweit sie für die Softwareerstellung relevant ist),</li> <li>... die unterschiedlichen Vertragsarten, die im Umfeld der Softwareerstellung zum Einsatz kommen,</li> <li>... die wichtigsten Verhandlungsstrategien bei der Verhandlung von Softwareverträgen,</li> <li>... rechtliche Aspekte im Bereich der IT-Kriminalität,</li> <li>... die Relevanz der Vorlesungsthemen in der Praxis der industriellen Softwareerstellung.</li> </ul>
Inhalt:	<p>Dieses Modul vermittelt aus konzeptioneller Sicht die Grundlagen der Wirtschaftsinformatik, wie sie für die Softwareerstellung relevant sind. Die Vorlesungsinhalte wurden aufgrund der 15-jährigen Erfahrung des Dozenten in der Praxis zusammengestellt, auf Basis aktueller Forschung und Lehre.</p> <p>Einleitung</p>

	Begriffsklärung, Grundlagen Softwareökonomie Management von Softwareprojekten Wertermittlung von Software Bepreisung von Software Software-Marketing Verhandlungen und Verträge Bilanzierung und Rechnungslegung IT-Kriminalität Schadensabwendung Zusammenfassung
Studien-/ Prüfungsleistung:	Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Anzahl der TeilnehmerInnen)
Medienformen:	Folien/ Tafel/ E-Learning
Literatur:	z.B. Buxmann, P.; Diefenbach, H.; Hess, T.: „Die Softwareindustrie. Ökonomische Prinzipien, Strategien, Perspektiven“, Heidelberg, 2008 Herzwurm, G.; Pietsch, W.: „Management von IT-Produkten“. Heidelberg, 2009 Mertens, P. (Hrsg.): „Lexikon der Wirtschaftsinformatik“, 3. Auflage, Heidelberg 2001 Versteegen, G.: „Marketing in der IT-Branche“. Heidelberg 2003 Wöhe, G.: „Einführung in die Allgemeine Betriebswirtschaftslehre“. München 2010

Modulbezeichnung:	<b>Wissensmanagement und Entscheidungen im Software Engineering (ISWKM)</b>
ggf. Modulniveau:	Bachelor / Master
ggf. Kürzel:	ISWKM
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dr. Privat-Doz. Andrea Herrmann
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung+Übung 2 SWS
Arbeitsaufwand:	60 h insgesamt, davon

	30 h Präsenzstudium 15 h Prüfungsvorbereitung 15 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)
Kreditpunkte:	2 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Vorlesung und Übung „Einführung in Software Engineering“ (ISW) oder vergleichbare Vorkenntnisse
Lernergebnisse:	Die Teilnehmer/innen kennen vertiefende Software Engineering Methoden, die Entscheidungen unterstützen bei Anforderungspriorisierung, Entwurf, Managemententscheidungen und Risikomanagement. Sie wissen, wie man im Arbeitsalltag Wissen verwaltet und haben eine Einführung in die Entscheidungstheorie erhalten.
Inhalt:	<p>Wissensmanagement</p> <ul style="list-style-type: none"> <li>Ontologien</li> <li>Rationale</li> <li>Re-engineering</li> <li>learning organization</li> </ul> <p>Entscheidungen</p> <ul style="list-style-type: none"> <li>Management-Entscheidungen, Business Case</li> <li>Risikomanagement</li> <li>Anforderungspriorisierung</li> <li>Entscheidungen im Entwurf: ATAM, SAAM, CBAM</li> <li>Entscheidungstheorie</li> <li>Entscheiden unter Ungewissheit</li> <li>Mathematical Economics</li> <li>Entscheidung mit mehreren Parteien: Harvard-Konzept, Verhandlungen</li> <li>Spieltheorie</li> <li>Fehlentscheidungen/ Decision Traps/ Biases</li> </ul>
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Hausaufgaben (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer Abschlussprüfung.
Medienformen:	Folien & Tafel
Literatur:	Raiffa, Howard; Richardson, John; Metcalfe, David: "Negotiation analysis - the science and art of collaborative decision making", Belknap, Cambridge, 2002

Modulbezeichnung:	<b>Qualitätsmanagement (ISWQM)</b>
ggf. Modulniveau:	
ggf. Kürzel:	ISWQM
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 4. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Barbara Paech
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 3 SWS, Übung 3SWS
Arbeitsaufwand:	240 h, davon <ul style="list-style-type: none"> <li>• 90 h Präsenzstudium</li> <li>• 15 h Prüfungsvorbereitung</li> <li>• 135 h Selbststudium und Aufgabenbearbeitung im Team</li> </ul>
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	Kenntnisse und Fähigkeiten wie sie in Modul ISW vermittelt werden
Empfohlene Vorkenntnisse:	Kenntnisse und Fähigkeiten wie sie in Modul ISW vermittelt werden
Lernergebnisse:	Dieses Modul vertieft die Grundkenntnisse und –fähigkeiten des Software Engineering insbesondere im Bereich des Qualitätsmanagement. Kenntnis der unter Inhalt angegebenen Methoden, Prozess und Werkzeuge Fähigkeit, Qualitätssicherung, Qualitätsmanagement und Prozessverbesserung und -management einem Softwareentwicklungsprojekt durchzuführen bzw unter Anleitung angewandte Forschung dazu durchzuführen. Fähigkeit Teilaufgaben im Team durchzuführen (eventuell mit „echten“ Kunden)
Inhalt:	Methoden, Prozesse und Werkzeuge für Qualitätssicherung Qualitätsmanagement Verbesserung von Softwareentwicklungsprozessen Management von Softwareentwicklungsprozessen
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (durch Erwerb einer entsprechenden Punktzahl) und erfolgreiche Teilnahme an der mündlichen oder schriftlichen Abschlussprüfung.
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	Wird jährlich aktualisiert

Modulbezeichnung:	<b>Requirements Engineering (ISWRE)</b>
ggf. Modulniveau:	
ggf. Kürzel:	ISWRE
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 4. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Barbara Paech
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 3 SWS, Übung 3SWS
Arbeitsaufwand:	240 h, davon <ul style="list-style-type: none"> <li>• 90 h Präsenzstudium</li> <li>• 15 h Prüfungsvorbereitung</li> <li>• 135 h Selbststudium und Aufgabenbearbeitung im Team</li> </ul>
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	Kenntnisse und Fähigkeiten wie sie in Modul ISW vermittelt werden
Empfohlene Vorkenntnisse:	Kenntnisse und Fähigkeiten wie sie in Modul ISW vermittelt werden
Lernergebnisse:	Dieses Modul vertieft die Grundkenntnisse und –fähigkeiten des Software Engineering insbesondere im Bereich des Requirements Engineering. Kenntnis der unter Inhalt angegebenen Methoden, Prozess und Werkzeuge Fähigkeit, Unternehmensmodellierung, Anforderungserhebung und –verhandlung und Requirements Management in einem Softwareentwicklungsprojekt durchzuführen bzw unter Anleitung angewandte Forschung dazu durchzuführen. Fähigkeit Teilaufgaben im Team durchzuführen (eventuell mit „echten“ Kunden)
Inhalt:	Methoden, Prozesse und Werkzeuge für Unternehmensmodellierung Prozessverbesserung in Unternehmen Anforderungserhebung und -verhandlung Requirements Management (Verbreitung, Prüfung und Aktualisierung von Anforderungen)
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (durch Erwerb einer entsprechenden Punktzahl) und erfolgreiche Teilnahme an der mündlichen oder schriftlichen Abschlussprüfung.
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	Wird jährlich aktualisiert

Modulbezeichnung:	<b>Visualisierung im Bereich Cultural Heritage (IVCH)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IVCH
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 4. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dr. Susanne Krömker
Sprache:	Englisch oder Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS
Arbeitsaufwand:	60 h; davon 30 h Präsenzstudium, 30 h Prüfungsvorbereitung
Kreditpunkte:	2 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Einführung in die Praktische Informatik (IPR), Programmierkurs (IPK), Computergraphik 1 (ICG1)
Lernergebnisse:	Die Studierenden sind mit unterschiedlichen Scantechniken vertraut, können Georadardaten interpretieren, sie wissen um die ethischen Grundsätze bei der Rekonstruktion, Befund und Hypothese (London Charter), kennen die Herangehensweise mit 2D und 3D Bildverarbeitung zur Erkennung von Merkmalen (Schrift).
Inhalt:	Weißlicht- und Time-of-flight-Scanner, Rekonstruktionen von Gefäßen und Gebäuden, 3D-Puzzle, Skelettierung, ethische Grundsätze
Studien-/ Prüfungsleistung:	Teilnahme an den Vorlesungen und Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Teilnehmerzahl)
Medienformen:	Folien / Lehrbuch
Literatur:	Clive Orton: Mathematics in Archaeology. Cambridge, MA, Cambridge University Press, 1982 Katsushi Ikeuchi, Daisuke Miyazaki (editors): Digitally Archiving Cultural Objects. Springer, 2007 Christian Hörr. Algorithmen zur automatisierten Dokumenten- tation und Klassifikation archäologischer Gefäße. Dissertation, TU Chemnitz, 2011

Modulbezeichnung:	<b>Verteilte Datenbanken und Informationssysteme (IVDB)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IVDB

ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 5. Semester Bachelor oder 1. Semester Master
Modulverantwortliche(r):	Studiendekan/Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Michael Gertz
Sprache:	Deutsch/Englisch nach Vereinbarung
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übungen 2 SWS
Arbeitsaufwand:	240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)
Kreditpunkte:	8 ECTS
Voraussetzungen nach Prüfungsordnung:	Keine
Empfohlene Vorkenntnisse:	Datenbanken 1 (IDB1), Datenbanken 2 (IDB2)
Lernergebnisse:	Die Studierenden sind in der Lage, eine verteilte Datenbank zu entwerfen und zu realisieren sind in der Lage, verschiedene Entwurfsstrategien für verteilte Datenbanken auf verschiedene Anwendungsszenarien anzuwenden kennen die Techniken und Prinzipien der verteilten Anfragebearbeitung und Anfrageoptimierung verstehen die Grundkonzepte und Protokolle zum Management verteilter Transaktionen, der Mehrbenutzersynchronisation und der Verwaltung von Datenreplikas wissen, welche Grundkonzepte und Techniken Multi-Datenbanken zugrunde liegen wissen, welche Techniken und Architekturen für die Datenverwaltung and Anfragebearbeitung im Rahmen verschiedener P2P Systeme existieren können die wichtigsten Konzepte, die den mobilen Datenbanken zugrunde liegen, darstellen kennen Verfahren zur verteilten Verwaltung von und Anfragebearbeitung auf Nicht-Standard Daten, wie z.B. geographische Daten oder große XML Datenmengen
Inhalt:	Grundbegriffe: verteilte Datenverarbeitung, Konzepte verteilter Datenbanken, Rechnernetzwerke Architekturen verteilter Datenbankmanagementsysteme (VDBMS): Transparenz, Client/Server Modelle Verteilter Datenbankentwurf: Entwurfsmethoden, Fragmentierung, Allokation, Integritätsbedingungen Anfragebearbeitung: Zielsetzungen, Relational



	<p>Algebra, Ebenen der Anfragebearbeitung, Anfrage-Dekomposition und Datenlokalisierung</p> <p>Anfrageoptimierung: zentrale Optimierung, Join Algorithmen, verteilte Optimierung von Anfragen (z.B. Ingres)</p> <p>Verteilte Transaktionsverwaltung und Mehrbenutzersynchronisation: Eigenschaften und Arten von Transaktionen, Serialisierbarkeit, Concurrency Control, Locking-Strategien</p> <p>Management und Verwaltung von Daten-Replikas</p> <p>Multi-Datenbanksysteme (MDBS): Probleme heterogener MDBS, Integrationsstrategien, Architekturen.</p> <p>Grundlagen des P2P Datenmanagement</p> <p>Nicht-Standard Systeme zur verteilten Datenverwaltung: verteiltes Information Retrieval, mobile Datenbanken, verteilte Geodatenbanken, Scientific Workflows, Grid-Datenbanken</p>
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	<p>z. B.: Tamer Ozsu and Patrick Valduriez: Principles of Distributed Database Systems, 2nd edition, Prentice Hall, 1999.</p> <p>Weitere Literatur wird vom Dozenten bekannt gegeben.</p>

Modulbezeichnung:	<b>Verteilte Systeme I (IVS1)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IVS1
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor oder 1.Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Artur Andrzejak und Mitarbeiter/in
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 2 SWS, Übung 2 SWS am Rechner mit Diskussion der Übungsaufgaben.
Arbeitsaufwand:	180 h
Kreditpunkte:	6 LP
Voraussetzungen nach Prüfungsordnung:	

Empfohlene Vorkenntnisse:	Kenntnisse in Java (z.B. durch Einführung in Software Engineering (ISW)) sowie in Betriebssystemen (z.B. IBN)
Lernergebnisse:	<p>Verständnis der unterschiedlichen parallelen Architekturen und Besonderheiten von verteilten Systemen</p> <p>Kenntnis der grundlegenden theoretischen Probleme und Algorithmen in verteilten Systemen (z.B. Datenaufteilung auf Prozessoren)</p> <p>Kenntnis über Anwendbarkeit diverser paralleler Frameworks (Pthreads, MPI, Map-Reduce) für eine gegebene Architektur</p> <p>Fähigkeit zur Erstellung von parallelen und verteilten Programmen in Java (z.B. mit Threads, RMI) oder Groovy (GPar)</p> <p>Fähigkeit der praktischen Verwendung von „modernen“ Ansätzen der parallelen/verteilten Programmierung (Map-Reduce, Actors, datenstromorientierte Ansätze)</p> <p>Vertrautheit mit Praxisfällen (Web-Dienste, parallele Verarbeitung von Daten)</p>
Inhalt:	<p>Das Modul behandelt die grundlegenden Prinzipien und Techniken der parallelen und verteilten Systeme im Kontext von Netzanwendungen und der parallelen Verarbeitung von Daten. Es werden Konzepte aus den Bereichen Architekturen, Protokolle, Programmierung, Softwareframeworks und Algorithmen vorgestellt. Ein wesentlicher Teil der Vorlesung widmet sich der parallelen und verteilten Programmierung, insbesondere moderneren Ansätzen wie Map-Reduce, Actors, datenstromorientierte Programmierung. Ergänzende Themen umfassen Fehlertoleranz, effiziente Protokolle und Skalierbarkeit. Die Umsetzung in die Praxis erfolgt an Beispielen aus den Bereichen Web-Dienste oder Verarbeitung großer Datenmengen. Das Modul soll die Studierenden befähigen, Spezifika und Probleme der verteilten Systeme zu verstehen, verteilte Anwendungen zu programmieren und Softwareframeworks wie Hadoop effizient einzusetzen.</p>
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer Abschlussprüfung.
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	<p>Andrew S. Tanenbaum, Maarten Van Steen: Distributed Systems: Principles and Paradigms, Prentice Hall, 2006;</p> <p>George Coulouris, Jean Dollimore, Tim Kindberg: Distributed Systems: Concepts and Design (4th ed.), Addison-Wesley, 2005.</p> <p>Kurose, James F., Keith W. Ross: Computernetzwerke : der Top-Down-Ansatz, Pearson Studium, 2008.</p> <p>Grama, A., Gupta, A., Karypis, G., Kumar V.: Intro-</p>

duction to Parallel Computing, Addison-Wesley, 2. Auflage, 2003.  
 Casanova, H.; Legrand, A.; Robert, Y.:Parallel Algorithms, Chapman & Hall/CRC, 2008.

Modulbezeichnung:	<b>Visualisierung in Natur- und Technikwissenschaften (IVNT)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IVNT
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Heike Leite
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS, Übung 1 SWS
Arbeitsaufwand:	120 h 45 h Präsenzstudium 15 h Prüfungsvorbereitung 60 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)
Kreditpunkte:	4 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Grundlagen der wissenschaftlichen Visualisierung
Lernergebnisse:	Die Studierenden kennen die Verfahren zur Aufnahme und Simulation von wissenschaftlichen Daten in den Natur- und Technikwissenschaften. kennen verschiedene Formate zur Speicherung wissenschaftlicher Daten und deren interne Repräsentation innerhalb von Visualisierungssoftware. sind mit der Volumenvisualisierung vertraut und können die Algorithmen selbständig implementieren. verstehen die wesentlichen Algorithmen zur effizienten Darstellung und Analyse von Vektorfeldern. kennen die Verfahren zur topologischen Analyse von wissenschaftlichen Felddaten. kennen Algorithmen zur visuellen Analyse von multivariaten Daten und Zeitreihen.
Inhalt:	Datenakquise und -repräsentation in Biologie und Medizin Volumenvisualisierung

	Datensimulation und -repräsentation in Umwelt- und Technikwissenschaften Effiziente Vektorfeldvisualisierung Topologische Verfahren in der Visualisierung Visualisierung von multivariaten Daten und Zeitreihen
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung
Medienformen:	Folien / Tafel / Skript / Lehrbuch
Literatur:	C. Hanson, C. Johnson: The Visualization Handbook, Elsevier, 2005.  R. Fernando: GPU Gems, NVidia Corp., 2004.  LaMothe: Tricks of the 3D Game Programming Gurus — Advanced 3D Graphics and Rasterization, Sams Publications, 2003.  Aktuelle Fachveröffentlichungen

Modulbezeichnung:	<b>Volumenvisualisierung (IVV)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IVV
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor, ab 1. Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Prof. Dr. Jürgen Hesser
Sprache:	Englisch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS, Übung 3 SWS
Arbeitsaufwand:	240 h 75 h Präsenzstudium 15 h Prüfungsvorbereitung 150 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen
Lernergebnisse:	Die Studierenden verstehen, wie experimentelle Daten zustande kommen und welche prinzipiellen Informationselemente diese enthalten, die für eine wissenschaftliche Visualisierung notwendig sind. Sie lernen die Prinzipien kennen, wie man zwischen kontinuierlichen (wirkliche Verteilung) und

	<p>diskreten Signalen (Darstellung im Computer) wechseln kann. Zudem werden sie eingeführt in Methoden der Konversion von Oberflächendaten in Volumendaten und umgekehrt. Schließlich werden ausgehend von theoretischen physikalischen Prinzipien sie in der Lage sein, verschiedene Visualisierungstechniken abzuleiten und die Näherungen, die hierfür gemacht werden, zu verstehen.</p> <p>Sie sind damit in der Lage, komplexe volumenorientierte Daten adäquat mathematisch zu repräsentieren, zu transformieren und die wesentlichen Strukturen mit adäquat darauf angepassten Verfahren der Visualisierung darzustellen.</p>
Inhalt:	<p>Einführung in die Visualisierung wissenschaftlicher Daten aus den Natur- und Lebenswissenschaften.</p> <p>Diskrete und kontinuierliche Repräsentation von Daten sowie numerische und computergraphische Methoden der Interpolation.</p> <p>Methoden der Konversion von Oberflächenrepräsentationen in Volumenrepräsentationen und umgekehrt, Optimierungstechniken für effiziente Algorithmen</p> <p>Ableitung und Varianten von Volumenvisualisierungstechniken</p> <p>Beschleunigungsverfahren und Parallelisierungsmethoden für Volumenvisualisierung</p> <p>Programmiertechnik: GPU-Programmierung</p>
Studien-/ Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung
Medienformen:	Folien/Lehrbuch
Literatur:	<p>z.B.</p> <p>Engel et al: Real-Time Volume Graphics  <a href="http://www.real-time-volume-graphics.org">www.real-time-volume-graphics.org</a>,          Schroeder et al: VTK Textbook  <a href="http://www.kitware.com/products/books/vtkbook.html">http://www.kitware.com/products/books/vtkbook.html</a></p>

Modulbezeichnung:	<b>Wissenschaftliches Arbeiten (IWA)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IWA
ggf. Lehrveranstaltungen:	
Studiensemester:	1. Semester Master
Modulverantwortliche(r):	Studiendekan/Studiendekanin der Informatik
Dozent(in):	Dozenten/Dozentinnen der Informatik
Sprache:	Deutsch

Zuordnung zum Curriculum:	
Lehrform/ SWS:	2 SWS
Arbeitsaufwand:	60 h; davon 30 h Präsenzstudium 30 h Selbststudium und praktische Übungen (eventuell in Gruppen)
Kreditpunkte:	2 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	
Lernergebnisse:	<p>Die Studierenden kennen die wichtigsten Literaturquellen der Informatik wissen, welche Tools und Techniken zur Verwaltung von Literatur existieren und wie diese verwendet werden</p> <p>sind in der Lage, wissenschaftliche Texte (z.B. aus Tagungsbänden oder Journals) kritisch zu lesen und zu bewerten</p> <p>kennen die fortgeschrittenen Techniken der Präsentation eines wissenschaftlichen Vortrags können strukturiert, verständlich, klar und schlüssig vor jedem Publikum präsentieren</p> <p>(Fachkompetenz) Kenntnis zentraler Erfolgsfaktoren, die für die Erstellung einer wissenschaftlichen Arbeit und die Durchführung einer Präsentation wichtig sind.</p> <p>(Methodenkompetenz) Kenntnis der Herangehensweise, einzelner Techniken und Hilfestellungen des wissenschaftlichen Arbeitens und der Erstellung von Präsentationen.</p>
Inhalt:	<p>Wesensmerkmale einer wissenschaftlichen Arbeit, Planung und Vorbereitung der Erstellung, Hilfsmittel der Informationsgewinnung, Literaturrecherche, formale und sprachliche Anforderungen</p> <p>Aufbau und Strukturierung der Arbeit, Zitierweise, Gestaltung von Darstellungen, Literaturverzeichnis, Anhang</p> <p>Vorbereitung einer Präsentation, Analyse der Vortragssituation, Strukturierung einer Präsentation, Gestaltung des Manuskriptes, sprachliche Anforderungen: Verständlichkeit, Artikulation, Modulation; Blickkontakt, Gestik, Mimik, Körperhaltung, Visualisierung, Medieneinsatz, Interaktion mit Zuhörern</p>
Studien-/ Prüfungsleistung:	Regelmäßige Teilnahme, Teilnahme an praktischen Übungen und Rollenspielen
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch

Literatur:	
------------	--

**Module, die auch im Bachelor/Master Mathematik angeboten werden:**

Modulbezeichnung:	<b>Mathematische Logik (MB9)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MB9
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3.Semester Bachelor
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Lineare Algebra I (MA4), Einführung in praktische Informatik (IPI)
Lernergebnisse:	Einführung in die verschiedenen Teilgebiete der Mathematischen Logik.
Inhalt:	Prädikatenlogik: Untersuchung der in der Mathematik üblichen logischen Schlussweisen. Mengenlehre: Grundagentheorie der Mathematik sowie Theorie der Ordinal- und Kardinalzahlen. Modelltheorie: Zusammenhang zwischen axiomatischen Theorien und ihren Modellen mit Beispielen aus der Algebra. Berechenbarkeitstheorie: Eigenschaften des Begriffes der berechenbaren Funktion. Beweistheorie: Grenzen der Formalisierbarkeit, Unvollständigkeit und Unentscheidbarkeit.
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitpunkt einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.
Medienformen:	
Literatur:	Bekanntgabe in der Vorlesung

Entnommen aus den Modulbeschreibungen BA Mathematik vom 23.01.2013

Modulbezeichnung:	<b>Wahrscheinlichkeitstheorie (MC4)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MC4
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 4. Semester Bachelor
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Analysis I und II (MA1, MA2) , Lineare Algebra I und II (MA4, MA5), Höhere Analysis (MA3), Einführung in die Wahrscheinlichkeitstheorie und Statistik (MA 8)
Lernergebnisse:	Grundlagen für alle Gebiete der Wahrscheinlichkeitstheorie und Statistik.
Inhalt:	<p>Maß- und Integrationstheorie: <math>\sigma</math>-Algebren, Borel-<math>\sigma</math>-Algebra, messbare Abbildungen, Konstruktion von Wahrscheinlichkeitsmaßen, Produkträume.</p> <p>Erwartungswert als Maßintegral, Sätze von Lebesgue, Beppo Levi, Fubini und Radon-Nikodym.</p> <p>Konvergenz von Zufallsvariablen: <math>L_p</math>-Räume, Zusammenhang zwischen fast sicherer, stochastischer und <math>L_p</math>-Konvergenz, Starkes Gesetz der großen Zahlen, Konvergenz in Verteilung, charakteristische Funktionen, zentraler Grenzwertsatz.</p> <p>Bedingte Verteilungen: Bedingte Erwartungen, Markov-Kerne, Martingale in diskreter Zeit.</p> <p>Stochastische Prozesse: Brownsche Bewegung, Poisson-Prozess, Empirischer Prozess.</p>
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur. Art und Zeitpunkt einer Wiederholungsprüfung wird vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.
Medienformen:	
Literatur:	<ul style="list-style-type: none"> <li>- Bauer, H.: Wahrscheinlichkeitstheorie, de Gruyter</li> <li>- Billingsley, P.: Probability and Measure, Wiley.</li> <li>- Dudley, R.N.: Real Analysis and Probability</li> <li>- Durrett, S.: Probability: Theory and Examples, Duxbury Press</li> </ul>



	- Jacod, J. and Protter, P.: Probability Essentials, Springer - Shiryayev, A.: Probability, Springer.
--	--

Entnommen aus den Modulbeschreibungen BA Mathematik vom 23.01.2013

Modulbezeichnung:	<b>Numerik (MD1)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MD1
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 3. Semester Bachelor
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240 h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Analysis 1 (MA1), Lineare Algebra 1 (MA4), Einführung in die Numerik (MA7)
Lernergebnisse:	Kenntnisse der numerischen Lösung von Anfangswert- und Randwertaufgaben gewöhnlicher Differentialgleichungen und einfacher partieller Differentialgleichungen.
Inhalt:	<p>Theorie von Anfangs- und Randwertaufgaben          Einschrittmethoden: Konsistenz, Stabilität, Konvergenz.          Numerische Stabilität und steife Anfangswertaufgaben          Andere Verfahrensklassen: Lineare Mehrschrittmethoden, Extrapolationsmethoden, Galerkin-Methoden (optional).          Lösung von Differentiellalgebraischen Aufgaben          Lösung von Randwertaufgaben: Schießverfahren, Differenzen- und Galerkin-Verfahren (optional).          Differenzenverfahren für elliptische partielle Differentialgleichungen, Laplace-Gleichung, 5-Punkte-Approximation.          Iterative Lösungsverfahren für diskretisierte Probleme.</p>
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteten 2-stündigen Klausuren, Wiederholungsmöglichkeit mit der Vorlesung im Folgejahr.
Medienformen:	
Literatur:	Bekanntgabe in der Vorlesung (Vorlesungsskriptum)

Modulbezeichnung:	<b>Statistik (MD2)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MD2
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 4. Semester Bachelor
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Analysis I (MA1), Lineare Algebra I (MA4), Einführung in die Wahrscheinlichkeitstheorie u. Statistik (MA8), Wahrscheinlichkeitstheorie (MC4)
Lernergebnisse:	Prinzipien der mathematischen Statistik
Inhalt:	Entscheidungstheorie: Dualität von Tests und Konfidenzbereichen, Neyman-Pearson-Theorie, allgemeine Entscheidungsverfahren, Risikofunktionen, Bayes- und Minimaxoptimalität Asymptotische Statistik: Verteilungsapproximation, Fisher-Information, relative asymptotische Effizienz von Tests und Schätzern, Likelihood-basierte Verfahren, nichtparametrische Verfahren.
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur. Art und Zeitpunkt einer Wiederholungsprüfung wird vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben
Medienformen:	
Literatur:	- Bickel, P. J. and Doksum, K. A.: Mathematical Statistics, Prentice Hall - Lehmann, E. L.: Testing Statistical Hypothesis, Springer Verlag - Van der Vaart, A. W.: Asymptotic Statistics, Cambridge University Press

Modulbezeichnung:	<b>Lineare Optimierung (MD3)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MD3
ggf.	

Lehrveranstaltungen:	
Studiensemester:	Ab 2.Semester Bachelor
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Lineare Algebra I, Programmierkenntnisse
Lernergebnisse:	Probleme, Theorie, Methoden und Algorithmen der Linearen Optimierung Selbständiges Lösen von Aufgaben aus dem Themenbereich mit Präsentation in den Übungen, Bearbeiten von praktischen Programmieraufgaben
Inhalt:	Formulierung von linearen Optimierungsproblemen Dualitätstheorie Struktur von Polyedern Die Simplexmethode, Grundversion und Varianten Der duale Simplex-Algorithmus Postoptimale Analyse und Re-Optimierung Polynomiale Algorithmen zur Linearen Optimierung Innere-Punkte-Methoden
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.
Medienformen:	
Literatur:	Padberg: Linear Optimization and Extensions Chvátal: Linear Programming Wright: Primal-Dual Interior-Point Methods

Entnommen aus den Modulbeschreibungen BA Mathematik vom 23.01.2013

Modulbezeichnung:	<b>Nichtlineare Optimierung (MD4)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MD4
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum	

Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Lineare Algebra I, Analysis I und II, Programmierkenntnisse
Lernergebnisse:	Grundlegende Methoden der Modellbildung und Simulation. Schwerpunktmäßig werden die Prozesse betrachtet, die sich mit Hilfe partieller Differentialgleichungen beschreiben lassen. Selbständiges Lösen von Aufgaben aus dem Themenbereich mit Präsentation in den Übungen, Bearbeiten von praktischen Programmieraufgaben
Inhalt:	Endlich-dimensionale, glatte, kontinuierliche, nichtlineare Optimierungsprobleme, Optimalitätsbedingungen für unbeschränkte und beschränkte Optimierungsprobleme, Gradientenverfahren, Konjugierte Gradienten-(CG-)Verfahren, Line Search, Newton und Quasi-Newton-SQP-Verfahren, Gauß-Newton-Verfahren, Behandlung von Ungleichungsbeschränkungen, Trust Region-Verfahren, Automatische Differentiation
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben. Benotete Klausur bzw. mündliche Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.
Medienformen:	
Literatur:	Nocedal, Wright: Numerical Optimization Gill, Murray, Saunders, Wright: Practical Optimization Geiger, Kanzow: Numerik (un)restringierter Optimierung Jarre, Stoer: Optimierung

Entnommen aus den Modulbeschreibungen BA Mathematik vom 23.01.2013

Modulbezeichnung:	<b>Wissenschaftliches Rechnen (MD5)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MD5
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	

Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Mathematische Grundvorlesungen (MA1-MA8)
Lernergebnisse:	Grundlegende Methoden der Modellbildung und Simulation. Schwerpunktmäßig werden die Prozesse betrachtet, die sich mit Hilfe partieller Differentialgleichungen beschreiben lassen.
Inhalt:	Hauptthemen sind Modellbildung: Modellierungssystematik, Diskrete Systeme, kontinuierliche Prozesse, Standardansätze zur Modellierung, Erhaltungsgleichung. Simulationsmethoden: Grundlegende Diskretisierungsverfahren, elementare Lösertechniken. Anwendungsbeispiele: Hier kommen einfache Anwendungen aus Biologie, Medizin, Physik u. a. zur Diskussion
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitpunkt einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.
Medienformen:	
Literatur:	Es wird ein Skriptum angeboten.

Entnommen aus den Modulbeschreibungen BA Mathematik vom 23.01.2013

Modulbezeichnung:	<b>Computeralgebra I (MG19)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MG19
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240 h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene	Algebra I (MB1)

Vorkenntnisse:	
Lernergebnisse:	Grundkenntnisse in Computeralgebra.
Inhalt:	<p>Die Vorlesung Computeralgebra befasst sich mit der Theorie und der Komplexität grundlegender mathematischer Algorithmen und deren Implementierungen in Computeralgebrasystemen. Hauptthemen sind:</p> <p><i>I. Schnelle Arithmetik:</i> Komplexität der elementaren Grundoperationen, diskrete Fouriertransformation, schnelle Multiplikation und schneller Euklidischer Algorithmus, Subresultanten und Polynomrestfolgen, modulare Algorithmen, Rechnen mit algebraischen Zahlen, schnelle Matrizenmultiplikation</p> <p><i>II. Primzerlegung und Primzahltests:</i> Primzahltest von Solovay- Strassen und Miller-Rabin, der AKS-Primzahlentest, RSA-Schema, elementare Primzahlzerlegungsverfahren, quadratisches Sieb, Irreduzibilitätstest für Polynome, Berlekamp- Algorithmen, Zassenhaus-Algorithmus, Gitter-Basis-Reduktion, Faktorisierung multivariater Polynome</p> <p><i>III. Gröbnerbasen-Algorithmen:</i> Gröbnerbasen und reduzierte Gröbnerbasen, Buchberger-Algorithmus, Eliminationstheorie, Algorithmen für elementare Idealoperationen, Berechnung der Dimension eines Ideals.</p>
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter Klausur- bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekanntgegeben.
Medienformen:	
Literatur:	<ul style="list-style-type: none"> <li>- J. von zur Gathen, J. Gerhard: Modern Computer Algebra</li> <li>- O. Geddes, S. R. Czapor, G. Labahn: Algorithms for Computer Algebra</li> <li>- D. Cox, J. Little, D. O'Shea: Ideals, Varieties and Algorithms</li> <li>- B. H. Matzat: Computeralgebra (Skriptum, in Vorbereitung)</li> </ul>

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Computeralgebra II (MG20)</b>
-------------------	----------------------------------

ggf. Modulniveau:	
ggf. Kürzel:	MG20
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Algebra I und II (MB1, MB2), Computeralgebra I (MG19)
Lernergebnisse:	Vertiefte Kenntnisse in Computeralgebra
Inhalt:	Die Vorlesung Computeralgebra II behandelt eines oder mehrere Gebiete aus dem folgenden Themenkatalog:  I. Algorithmische Zahlentheorie II. Algorithmische kommutative Algebra III. Algorithmische Gruppentheorie IV. Algorithmische Invariantentheorie V. Algorithmische Arithmetische Geometrie
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter Klausur- bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.
Medienformen:	
Literatur:	Wird in der Vorlesung bekannt gegeben

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Numerische Lineare Algebra (MH5)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MH5
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	

Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Einführung in die Numerik (MA7)
Lernergebnisse:	Kenntnis der gebräuchlichen Methoden zur numerischen Lösung von Aufgaben der Linearen Algebra
Inhalt:	Lineare Gleichungssysteme und Eigenwertaufgaben Iterative Verfahren, Fixpunktiterationen Krylowraum-Methoden Iterative Verfahren für Eigenwertaufgaben Singularwertzerlegung Anwendungen auf Systemmatrizen bei der Diskretisierung partieller Differentialgleichungen
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur (und einer Freischussmöglichkeit), Wiederholungsmöglichkeit mit der Vorlesung in den Folgejahren.
Medienformen:	
Literatur:	Bekanntgabe in der Vorlesung (Vorlesungsskriptum)

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Numerik partieller Differentialgleichungen (MH7)</b>
ggf. Modulniveau:	
ggf. Kürzel:	
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Analysis I (MA1), Lineare Algebra I (MA4), Einführung in die Numerik (MA7), Partielle Differentialgleichungen I (MC2)
Lernergebnisse:	Kenntnis der gebräuchlichen Methoden zur



	numerischen Lösung von Anfangswert- und Anfangs-Randwert-Aufgaben der prototypischen partiellen Differentialgleichungen: Laplace-, Wärmeleitungs- und Wellengleichung.
Inhalt:	I. Theorie von partiellen Differentialgleichungen, Typeneinteilung und Lösungseigenschaften II. Differenzenverfahren für elliptische Randwertaufgaben III. Finite-Elemente-Verfahren für elliptische Randwertaufgaben: Diskretisierung, a posteriori Fehlerschätzung und Gitteradaption IV. Lösungsverfahren für große lineare Gleichungssysteme: Fixpunktiterationen, Krylowraum- und Mehrgittermethoden V. Verfahren für parabolische Anfangs-Randwertaufgaben (Wärmeleitungsgleichung) VI. Verfahren für hyperbolische Probleme (Wellengleichung)
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur (und einer Freischussmöglichkeit), Wiederholungsmöglichkeit mit der Vorlesung in den Folgejahren.
Medienformen:	
Literatur:	Bekanntgabe in der Vorlesung (Vorlesungsskriptum)

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Numerische Optimierung bei Differentialgleichungen (MH8)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MH8
ggf. Lehrveranstaltungen:	
Studiensemester:	Studiendekan Mathematik
Modulverantwortliche(r):	
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Kenntnisse, wie sie in den Vorlesungen Einführung in die Numerische Mathematik und Numerische Mathematik I vermittelt werden sowie Grundkenntnisse der linearen Algebra und Analysis sowie einer

	Programmiersprache, Vorlesung Gewöhnliche Differentialgleichungen (MC1)
Lernergebnisse:	Parameterschätzung sowie optimale nichtlineare Versuchsplanung bei Differentialgleichungen.
Inhalt:	Das Modul behandelt Grundlagen und numerische Methoden der optimalen Steuerung.
Studien-/Prüfungsleistung:	Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer Abschlussprüfung.
Medienformen:	
Literatur:	

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Statistik II (MH12)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MH12
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	Analysis I (MA1), Lineare Algebra I (MA4)
Empfohlene Vorkenntnisse:	Wahrscheinlichkeitstheorie I (MC5), Statistik I (MD3)
Lernergebnisse:	Vertiefte Behandlung einer Auswahl statistischer Methoden
Inhalt:	<p>Mögliche Themen sind:</p> <p><i>I. Multivariate Statistik:</i> Wishart-Verteilung, multipler Korrelationskoeffizient, Hotellings T<sup>2</sup>-Verteilung, Hauptkomponentenanalyse, kanonische Korrelationen, grafische Modelle</p> <p><i>II. Zeitreihenanalyse:</i> Lineare Filter, ARMA-Modelle, Prädiktion, State-Space Modelle, Spektraldarstellung, Periodogramm, Whittle-Likelihood, nichtlineare Zeitreihenmodelle</p> <p><i>III. Nichtparametrik:</i> Dichteschätzung und nichtparametrische Regression, Kernschätzer, lokal polynomiale Schätzer, Orthogonalreihen-schätzer,</p>

	Adaptivität, Risikoabschätzung, nichtparametrische Tests
Studien-/Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur. Art und Zeitrahmen einer Wiederholungsprüfung wird vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.
Medienformen:	
Literatur:	<ul style="list-style-type: none"> <li>- Anderson, T. W.: An Introduction to Multivariate Statistical Analysis, John Wiley</li> <li>- Jørgensen, Bent: The Theory of Linear Models, Chapman &amp; Hall, New York, 1993.</li> <li>- Brockwell, P. J. and Davis R. A.: Time Series: Theory and Methods, Springer-Verlag</li> <li>- Wasserman, L.: All of Nonparametric Statistics, Springer-Verlag</li> </ul>

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Wahrscheinlichkeitstheorie II (MH13)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MH13
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Analysis I (MA1), Lineare Algebra I (MA4), Wahrscheinlichkeitstheorie I (MC4)
Lernergebnisse:	Ausgewählte Themen zu Stochastischen Prozessen und zur Stochastischen Analyse.
Inhalt:	<p><i>I. Theorie Stochastischer Prozesse:</i> Endlich-dimensionale Verteilungen, Existenzsatz von Kolmogorov, stetige Pfade, Konstruktion und Eigenschaften der Brownschen Bewegung, Gaußprozesse</p> <p><i>II. Ergodentheorie:</i> Stationäre und ergodische Prozesse, Ergodensätze</p> <p><i>III. Invarianzprinzipien:</i> Straffheit, schwache</p>

	Konvergenz im Raum der stetigen Funktionen, Invarianzprinzip von Donsker, Theorie der Empirischen Prozesse  <i>IV. Stochastisches Integral: Martingale in stetiger Zeit, Itô-Integral, Itô-Formel</i>
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung wird vom Dozent festgelegt und zu Beginn der Vorlesung bekannt gegeben.
Medienformen:	
Literatur:	- Durrett, S.: Probability: Theory and Examples, Duxbury Press - Karlin, S. and Taylor, H.: A First/Second Course in Stochastic Processes, Academic Press  - Karatzas, I. and Shreve, S.: Brownian Motion and Stochastic Calculus, Springer

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Berechenbarkeit und Komplexität I (MH14)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MH14
ggf. Lehrveranstaltungen:	
Studiensemester:	Studiendekan Mathematik
Modulverantwortliche(r):	
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übungen 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Grundkenntnisse aus der Theoretischen Informatik sind hilfreich
Lernergebnisse:	Grundkenntnisse über Berechenbarkeit und Komplexität
Inhalt:	Die Berechenbarkeitstheorie liefert den formalen Rahmen, die Lösbarkeit algorithmischer Probleme zu untersuchen, die Komplexitätstheorie stellt Methoden und Konzepte zur Analyse des erforderlichen Aufwands algorithmischer Problemlösungen zur Verfügung. Ziel des Moduls ist es die Studierenden mit den zentralen Konzepten und Methoden der

	Berechenbarkeits- und der Komplexitätstheorie vertraut zu machen. In der Berechenbarkeitstheorie stehen Methoden zum Nachweis der Unentscheidbarkeit im Mittelpunkt, in der Komplexitätstheorie liegt der Schwerpunkt auf dem Vergleich und der strukturellen Analyse der polynomiell beschränkten Komplexitätsklassen. Insbesondere werden das P-NP-Problem und die NP-Vollständigkeit behandelt.
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.
Medienformen:	
Literatur:	

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Berechenbarkeit und Komplexität II (MH15)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MH15
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 3 SWS, Übungen 1 SWS
Arbeitsaufwand:	180 h
Kreditpunkte:	6 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Berechenbarkeit und Komplexität I (MH14)
Lernergebnisse:	Vertiefte Kenntnisse über Berechenbarkeit und Komplexität
Inhalt:	In diesem Modul werden ausgewählte fortgeschrittene Themen aus dem Bereich der Berechenbarkeits- und Komplexitätstheorie behandelt.
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben

	.
Medienformen:	
Literatur:	

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Algorithmische Optimierung I (MH16)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MH16
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240 h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Mathematische Grundvorlesungen (MA1- MA8)
Lernergebnisse:	Grundkenntnisse über algorithmische Optimierung
Inhalt:	Das Modul behandelt moderne Verfahren der unbeschränkten und beschränkten Optimierung. Die Studierenden werden in die Lage versetzt, moderne Verfahren des Gebietes anzuwenden, zu beurteilen und zu entwickeln.
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.
Medienformen:	
Literatur:	Bekanntgabe in der Vorlesung

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Algorithmische Optimierung II (MH17)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MH17
ggf. Lehrveranstaltungen:	
Studiensemester:	

Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Algorithmische Optimierung I (MH16)
Lernergebnisse:	Grundlagen der linearen und ganzzahligen Optimierung.
Inhalt:	I. Dualitätstheorie II. Simplexalgorithmus und Varianten III. Innere-Punkte-Verfahren IV. Schnittebenen-Verfahren
Studien-/ Prüfungsleistung:	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.
Medienformen:	
Literatur:	Bekanntgabe in der Vorlesung

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Mustererkennung (MH18)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MH18
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Mathematische Grundvorlesungen (MA1- MA8), Wahrscheinlichkeitstheorie I (MC4), Numerische Lineare Algebra (MH5), Algorithmische Optimierung I

	(MH16)
Lernergebnisse:	Selbständiges computergestütztes Lösen von Problemen der Statistischen Mustererkennung.
Inhalt:	Euklidische Einbettungen, Multidimensionale Skalierung, Bayes Klassifikator, Fehlerschranken und -abschätzungen, Entscheidungsbäume, Kombination und Performanzsteigerung einfacher Klassifikatoren, Klassifikation mit Kernfunktionen, Gaußsche Prozesse und Klassifikation, Klassifikation mit Mischungsverteilungen, nichtparametrische Klassifikation, Merkmals-auswahl und -extraktion, Ballungsanalyse mit Prototypen, Ähnlichkeitsgraphen und unüberwachtes Lernen.
Studien-/ Prüfungsleistung:	Bearbeiten von Übungsblättern und Übungen am Computer, Semesterbegleitende Prüfung, Art und Zeitrahmen werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben
Medienformen:	
Literatur:	Bekanntgabe in der Vorlesung und auf der entsprechenden WWW-Seite

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Bildverarbeitung (MH19)</b>
ggf. Modulniveau:	
ggf. Kürzel:	MH19
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum:	
Lehrform/SWS:	Vorlesung 4 SWS, Übung 2 SWS
Arbeitsaufwand:	240h
Kreditpunkte:	8 LP
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Mathematische Grundvorlesungen (MA1- MA8), Fouriertransformation, Variationsrechnung, Numerische Lineare Algebra (MH5), Partielle Differentialgleichungen II (MH3), Numerik Partieller Differentialgleichungen (MH7), Statistische Mustererkennung (MH18)
Lernergebnisse:	Selbständiges computergestütztes Lösen von Problemen der Digitalen Bildverarbeitung.



Inhalt:	Repräsentation der Bildfunktion (Fouriertransformation, Abtasttheorem, Wavelettransformation), Variationsmethoden zur Bildrestauration, Bildmerkmale, Textur, Optischer Fluss und Korrespondenzanalyse, Bildsegmentierung, Epipolareometrie und Stereorekonstruktion, Objekt- und Ereigniserkennung
Studien-/ Prüfungsleistung:	Bearbeiten von Übungsblättern und Übungen am Computer, Semesterbegleitende Prüfung, Art und Zeitrahmen werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben
Medienformen:	
Literatur:	Bekanntgabe in der Vorlesung und auf der entsprechenden WWW-Seite

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

Modulbezeichnung:	<b>Modellierung und Optimierung in Robotik und Biomechanik (MH24)</b>
ggf. Modulniveau:	Bachelor / Master
ggf. Kürzel:	MH24
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Mathematik
Dozent(in):	Prof. Dr. Katja Mombaur
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung 2 SWS, Übung 2 SWS
Arbeitsaufwand:	180 h
Kreditpunkte:	6 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Notwendig: Programmierkenntnisse in C/C++, Einführung in die Numerik Vorteilhaft: Algorithmische Optimierung 1, Numerik1, Kenntnisse in Matlab/Octave
Lernergebnisse:	Grundkenntnisse in der Modellierung und Optimierung von Bewegungen im Bereich Biomechanik und Robotik. Grundfertigkeiten im Umgang mit Softwarewerkzeugen für Modellierung, Visualisierung, Simulation und

	<p>Optimale Steuerung.  Selbständiges Lösen von einfachen mechanischen Modellierungsaufgaben und  Optimalsteuerungsaufgaben am Computer</p>
Inhalt:	<p>Inhalt der Vorlesung:  Dynamische Prozessmodellierung  Mechanische Grundbegriffe, Kinematik, Dynamik  Modellierung von Mehrkörpersystemen  Bewegungssimulation  Nichtlineare Optimierung  Direkte Methoden der Optimalen Steuerung  Steuerung und Regelung von Bewegungen  Modellierung menschlicher Geh- und  Rennbewegungen  Modellierung von Laufbewegungen humanoider und  zweibeiniger Roboter  Stabilität von Bewegungen</p> <p>Inhalt der Computerübungen:  Simulation und Visualisierung mechanischer Systeme  Modellierung der Dynamik von Mehrkörpersystemen  mit der RBDL (Rigid Body Dynamics Library)  Implementierung und Lösen von  Optimalsteuerungsproblemen mit MUSCOD-II</p>
Studien-/ Prüfungsleistung:	<p>Bestehen einer 2-stündigen benoteten Klausur.  Zulassungsvoraussetzung: erfolgreiche Teilnahme an  den Computerübungen (Anwesenheit, Lösung und  Präsentation von Programmieraufgaben).</p>
Medienformen:	<p>PowerPoint und Tafel /Tablett-Anschrieb  Computerübungen</p>
Literatur:	<p>D. Greenwood: Principles of Dynamics, Prentice Hall,  1987  I .Newton: Principia, 1687  J. T. Betts: Practical Methods for Optimal Control  Using  Nonlinear Programming. SIAM, Philadelphia, 2001  J. Craig: Introduction to Robotics - Mechanics and  Control. Prentice Hall, 2003  J. Nocedal, S. Wright: Numerical Optimization,  Springer, 2000  B. Siciliano, et al: Robotics - Modeling, Planning and  Control, Springer 2008  Spong, Hutchinson, Vidyasagar: Robot modeling and  control, Wiley, 2005  Perry, Burnfield: Gait Analysis - Normal and  pathological function, Slack Inc., 2010  M. Raibert: Legged Robots that Balance, MIT Press,  2000</p>

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 23.01.2013

### **Module, die im Master Technische Informatik angeboten werden:**

Alle fachlichen Module (nicht Fachübergreifende Kompetenzen) aus dem Master Technische Informatik können – entsprechend der inhaltlichen Voraussetzungen – auch im Master Angewandte Informatik belegt werden. Das Angebot findet sich im jeweils gültigen Modulhandbuch des Masters Technische Informatik.

[http://www.ziti.uniheidelberg.de/ziti/index.php?option=com\\_content&view=article&id=135&Itemid=77&lang=de](http://www.ziti.uniheidelberg.de/ziti/index.php?option=com_content&view=article&id=135&Itemid=77&lang=de)

### **Module, die auch im Bachelor/ Master Physik angeboten werden**

Modulbezeichnung:	<b>Physics of Imaging (MWInf5)</b>
ggf. Modulniveau	
ggf. Kürzel	MWInf5
ggf. Untertitel	
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Physik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum	
Lehrform/ SWS:	Lecture on „Physics of Imaging“ (4hours/week)
Arbeitsaufwand:	120 h
Kreditpunkte:	4 LP
Voraussetzungen nach Prüfungsordnung	
Empfohlene Voraussetzungen:	UKInf2, PEP1-PEP4
Lernergebnisse:	Basics of the Physics of Imaging; common principles and techniques of imaging for atomic to astronomical scales.
Inhalt:	<ul style="list-style-type: none"><li>-Projective geometry, optics, wave optics, Fourier optics and lens aberrations</li><li>-Radiometry of imaging</li><li>-Methods of imaging: scanning electron microscopy, X-ray, EDX, FLIM, FRET, fluorescence imaging, near-field imaging</li><li>-CCD and CMOS technology</li><li>-Holography, ultrasound imaging, CT- computer tomography, magnetic resonance imaging</li><li>-Satellite imaging, synthetic aperture radar, radio astronomy</li></ul>

Studien-/ Prüfungsleistungen:	Defined by lecturer before beginning of course
Medienformen:	
Literatur:	Announced by lecturer

Entnommen aus dem Modulhandbuch MScPhysik, Version 2012.2/v3.

Modulbezeichnung:	<b>Image Processing (MWInf6)</b>
ggf. Modulniveau	
ggf. Kürzel	MWInf6
ggf. Untertitel	
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Physik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum	
Lehrform/ SWS:	Lecture on „Image Processing“ (4hours/week) Exercises (2hours/week)
Arbeitsaufwand:	210 h
Kreditpunkte:	7 LP
Voraussetzungen nach Prüfungsordnung	
Empfohlene Voraussetzungen:	UKInf1
Lernergebnisse:	Learn how to analyze signals from time series, images, and any kind of multidimensional signals and to apply it to problems in natural sciences, life sciences and technology.
Inhalt:	<ul style="list-style-type: none"> <li>-Continuous and discrete signals, sampling theorem, signal representation</li> <li>-Fourier transform</li> <li>-Random variables and fields, probability density functions, error propagation</li> <li>-Homogeneous and inhomogeneous point operations</li> <li>-Neighbourhood operations, linear and nonlinear filters, linear system theory</li> <li>-Geometric transformations and interpolation</li> <li>-Multi-grid signal presentation and processing</li> <li>-Averaging, edge and line detection, local structure analysis, local phase and wave numbers</li> <li>-Motion analysis in image sequences</li> <li>-Segmentation</li> <li>-Regression, globally optimal signal analysis, variation approaches, steerable and nonlinear filtering, inverse filtering</li> </ul>

	-Morphology and shape analysis, moments, Fourier descriptors -Bayesian image restoration --Object detection and recognition
Studien-/Prüfungsleistungen:	Defined by lecturer before beginning of course
Medienformen:	
Literatur:	B. Jähne, Digital Image Processing, 6th edition, Springer

Entnommen aus dem Modulhandbuch MScPhysik, Version 2012.2/v3

Modulbezeichnung:	<b>Pattern Recognition (MWInf7)</b>
ggf. Modulniveau	
ggf. Kürzel	MWInf7
ggf. Untertitel	
ggf. Lehrveranstaltungen:	
Studiensemester:	
Modulverantwortliche(r):	Studiendekan Physik
Dozent(in):	
Sprache:	
Zuordnung zum Curriculum	
Lehrform/ SWS:	Lecture on „Pattern Recognition “ (4hours/week) Exercises (2hours/week)
Arbeitsaufwand:	210h
Kreditpunkte:	7 LP
Voraussetzungen nach Prüfungsordnung	
Empfohlene Voraussetzungen:	UKInf1; Knowledge about Linear Algebra, Probability, Statistics
Angestrebte Lernergebnisse:	Given a huge bunch of data, find out what's in it; build automated diagnostic systems or expert systems that automatically learn to make reliable predictions from a training set of examples. Lectures and exercises will be interwoven and allow you to build such systems by yourself; real-life examples will be drawn from for the application areas named below.
Inhalt:	-Curse of dimensionality -Variable selection and dimension reduction for high-dimensional data -Unsupervised learning: Cluster analysis -Supervised learning: Regression -Supervised learning: Classification by means of neural networks, support vector machines, etc.

	-Graphical models -Applications: Data mining, industrial quality control, process monitoring, astrophysics, medicine, life sciences
Studien-/Prüfungsleistungen:	Defined by lecturer before beginning of course
Medienformen:	
Literatur:	Pattern Classification (2nd ed.) by Richard O. Duda, Peter E. Hart. and David G. Stork. Wiley, 2000.

Entnommen aus dem Modulhandbuch MScPhysik, Version 2012.2/v3

Modulbezeichnung:	<b>Introduction to Image Processing on the GPU (IGPU)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IGPU
ggf. Lehrveranstaltungen:	
Studiensemester:	From 5th semester bachelor, or 1st semester master
Modulverantwortliche(r):	
Dozent(in):	Dr. Bastian Goldluecke
Sprache:	English course material and exercise sheets Course language English, switching to German only if there is no demand for English
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Four day block course, equivalent of 2 SWS
Arbeitsaufwand:	60 h 30 h Lectures and lab exercises 30 h Preparation and home exercises
Kreditpunkte:	2 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Solid knowledge of C programming, in particular no fear of pointers and direct memory access Some basics of image processing are helpful, but not strictly necessary
Lernergebnisse:	For certain computation tasks which are colloquially called „embarassingly parallel“, and which occur quite frequently for example in image analysis, an implementation on a GPU (i.e. graphics card) can be orders of magnitude faster than a similar CPU implementation. In this short course, the students learn:

	<ul style="list-style-type: none"> <li>- what kind of code can and can not be parallelized on a graphics processing unit (GPU)</li> <li>- the structure of a GPU, the different types of memory and how and when to use them</li> <li>- how to program the GPU using nVidia CUDA</li> <li>- how to perform various image analysis tasks using the GPU</li> <li>- ways to optimize the speed of GPU code</li> </ul>
Inhalt:	<p>The four days will be broken up into theoretical and practical segments. In the (short) theoretical lectures, I will introduce the concepts necessary for GPU programming and basic image analysis, which you will immediately try out afterwards in lab sessions. This way, the course will be highly practical and interactive. Students will work in small groups, each of which will have the task to write a small program solving an image analysis problem of their choice.</p> <p>Note that while image analysis tasks are a focus of the course because they give immediately visible results, the acquired techniques can be of course be employed in other fields as well, e.g. for solving large linear algebra problems or PDEs.</p>
Studien-/Prüfungsleistung:	Active participation in the sessions and completion of the assigned program.
Medienformen:	Slides / blackboard / online manuals / example code
Literatur:	e.g. nVidia CUDA Programming Guide, available in the "CUDA zone" on <a href="http://www.nvidia.com">www.nvidia.com</a>

Modulbezeichnung:	<b>Fast Parallel Implementations of Image Labeling Problems (IMLP)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IMLP
ggf. Lehrveranstaltungen:	
Studiensemester:	From 5th semester bachelor, or 1st semester master
Modulverantwortliche(r):	
Dozent(in):	Dr. Bastian Goldluecke
Sprache:	English course material and exercise sheets Course language English, switching to German only if there is no demand for English
Zuordnung zum Curriculum:	

Lehrform/ SWS:	2 SWS Lecture + 1 SWS Exercise
Arbeitsaufwand:	120 h 45 h Lectures and lab exercises 60 h Revision and home exercises 15 h Exam preparation
Kreditpunkte:	4 LP
Voraussetzungen nach Prüfungsordnung:	
Empfohlene Vorkenntnisse:	Either „Introduction to Image Processing on the GPU“ offered at the beginning of the semester, or an introduction to <i>Variational Image Analysis</i> from e.g. last semester.
Lernergebnisse:	<i>Image labeling problems</i> are a fundamental class of problems appearing in image analysis, which dominate tasks in low-level computer vision like depth and motion estimation. Recently, many algorithms have been developed to solve this kind of problems in a variational framework, which allows for fast parallel implementations on the GPU. In this short course, the students learn: - Theoretical background for solving labeling problems in a variational framework - Efficient algorithms to solve the related class of optimization problems with parallel algorithms, which can be implemented on the GPU - How to implement these algorithms using nVidia CUDA - Techniques and tricks to make the implementations efficient
Inhalt:	The lecture is aimed at students who either participated in my CUDA course at the beginning of the semester and want to learn more about image analysis and the theoretical background, or participated in a theoretical course on variational image analysis and want to learn more about state-of-the art labeling algorithms and the practical side of their implementation. If you are new to both topics, you might still take part in the course, but must be prepared to take (potentially a lot of) additional time learning the prerequisites.
Studien-/ Prüfungsleistung:	Active participation in lecture and exercises, oral exam.
Medienformen:	Slides / blackboard / online manuals / example code
Literatur:	e.g. Chambolle et al. 2010 <i>“An Introduction to Total Variation for Image Analysis”</i> for theoretical background, and nVidia CUDA Programming



	Guide, available in the “CUDA zone” on <a href="http://www.nvidia.com">www.nvidia.com</a> , for the practical one.
--	--

**Module, die auch im Bachelor/ Master Biologie angeboten werden**

Modulbezeichnung	<b>Model-based segmentation of biomedical images (Modellbasierte Segmentierung Biomedizinischer Bilder, IMSBI)</b>
Modulniveau	
Kürzel	IMSBI
Lehrveranstaltungen	
Studiensemester	
Modulverantwortlicher	Studiendekan/Studiendekanin der Informatik
Dozent	Dr. Stefan Wörz
Sprache	Deutsch oder Englisch
Zuordnung zum Curriculum	
Lehrform / SWS	Vorlesung 2 SWS
Arbeitsaufwand	120h insgesamt, davon 30h Präsenzstudium 30h Prüfungsvorbereitung 60h Selbststudium und Aufgabenbearbeitung
Kreditpunkte	4 LP
Voraussetzungen nach Prüfungsordnung	
Empfohlene Vorkenntnisse	Grundkenntnisse Signal- oder Bildverarbeitung (z.B. Vorlesung „Bildverarbeitung“ oder „Bioinformatik“)
Lernergebnisse	Ziel der Vorlesung ist es, einen Überblick über 2D und 3D deformierbare Modelle und deren mathematische Grundlagen für die modellbasierte Segmentierung biomedizinischer Bilder zu geben
Inhalt	In der Vorlesung werden drei wesentliche Klassen von deformierbaren Modellen (aktive Konturen, statistische Formmodelle, analytische parametrische Modelle) sowie starre Template Modelle behandelt. Die verschiedenen Segmentierungsverfahren werden an aktuellen Beispielen aus der 2D und 3D biomedizinischen Bildverarbeitung veranschaulicht. Geplante Themen sind unter anderem: - Introduction and overview of image analysis, image segmentation, and biomedical images - Template models, linear transformations, Hough transform

	<ul style="list-style-type: none"> <li>- Active contour models: snakes, level sets, and geodesic active contours</li> <li>- Statistical shape models: active shape models, active appearance models, and extensions</li> <li>- Shape correspondence estimation: landmark-based, intensity-based, and hybrid</li> <li>- Analytic parametric models: contour-based and intensity-based</li> <li>- Performance evaluation and applications</li> </ul>
Studien-/ Prüfungsleistung	Bestehen einer Abschlussprüfung
Medienformen	Folien und Tafel
Literatur	Bekanntgabe in der Vorlesung

### **Module im Anwendungsgebiet**

Modulbezeichnung:	<b>Anwendungsgebiet (IAG)</b>
ggf. Modulniveau:	
ggf. Kürzel:	IAG
ggf. Lehrveranstaltungen:	
Studiensemester:	Ab 1.Semester Master
Modulverantwortliche(r):	Studiendekan/ Studiendekanin der Informatik
Dozent(in):	Dozenten/ Dozentinnen der Informatik
Sprache:	Deutsch
Zuordnung zum Curriculum:	
Lehrform/ SWS:	Vorlesung, Übung und/oder Praktikum
Arbeitsaufwand:	540 h, Aufteilung in Präsenz- / Übungs- und Praktikumszeit in Absprache mit den Dozentinnen bzw. Dozenten
Kreditpunkte:	18 LP, davon 8 LP mindestens durch einen nicht-informatischen Modul auf Masterniveau (Ausnahmegenehmigung möglich), die restlichen 10 LP können durch ein Projekt oder Module auf Bachelorniveau erbracht werden
Voraussetzungen nach Prüfungsordnung:	keine
Empfohlene Vorkenntnisse:	Gleiches Anwendungsgebiet im Bachelor
Lernergebnisse:	Vertiefte Kenntnisse und Fähigkeiten in einem Anwendungsgebiet
Inhalt:	Wahl eines Anwendungsgebietes nach den Regeln der Prüfungsordnung Festlegung von und Teilnahme an Modulen aus dem Anwendungsgebiet (die LP entsprechen dabei den

	<p>Vorgaben aus dem Anwendungsgebiet). Dabei ist sicherzustellen, dass keine Module aus dem Anwendungsgebiet gewählt werden, die schon im Bachelor-Studium eingebracht wurden.</p> <p>(optional) Definition und Durchführung eines interdisziplinären Projektes, d.h.</p> <p>Festlegung eines Dozenten / einer Dozentin aus dem Anwendungsgebiet und der Informatik</p> <p>Gemeinsame Festlegung des Projektziels durch die Dozentinnen bzw. Dozenten und den/die Studierende.</p> <p>Das Projektziel umfasst eine informatische Leistung im Anwendungsgebiet.</p> <p>Festlegung des Arbeitsaufwandes und damit der LP für das Projekt</p> <p>Durchführung des Projekts entsprechend des Projektziels unter Anleitung der Dozentinnen bzw. Dozenten</p> <p>Dokumentation des Ergebnisses</p> <p>Erstellung eines Projektbericht</p> <p>Präsentation des Ergebnisses</p>
Studien-/ Prüfungsleistung:	<p>Prüfungsleistungen in dem Anwendungsgebiet und (optional) Prüfungsleistungen für das interdisziplinäre Projekt analog zum Modul IFP, gewichtet nach dem jeweiligen Anteil der LP</p>
Medienformen:	Folien/ Tafel/ E-Learning/ Lehrbuch
Literatur:	