

Modulhandbuch
Master-Studiengang
„Angewandte Informatik“

Ruprecht-Karls-Universität Heidelberg
Fakultät für Mathematik und Informatik

Fassung vom 14.07.2021 zur Prüfungsordnung vom 22.07.2010
(mit letzter Änderung vom 07.02.2013)

Studienform: Vollzeit

Art des Studiengangs: Konsekutiv

Regelstudienzeit: 4 Semester

Anzahl der im Studiengang zu erwerbenden Leistungspunkte: 120

Studienstandort: Heidelberg

Anzahl der Studienplätze: Keine Zulassungsbeschränkung

Gebühren/Beiträge: Gemäß allgemeiner Regelung der Universität Heidelberg

Präambel

Einordnung und Gesamtdarstellung des Master-Studiengangs Angewandte Informatik

Anknüpfend an ihr Leitbild und ihre Grundordnung verfolgt die Universität Heidelberg in ihren Studiengängen fachliche, fachübergreifende und berufsfeldbezogene Ziele in der umfassenden akademischen Bildung und für eine spätere berufliche Tätigkeit ihrer Studierenden. Das daraus folgende Kompetenzprofil wird als für alle Disziplinen gültiges Qualifikationsprofil in den Modulhandbüchern aufgenommen und in den spezifischen Qualifikationszielen sowie dem Curriculum und Modulen des Master-Studiengangs Angewandte Informatik umgesetzt:

- Entwicklung von fachlichen Kompetenzen mit ausgeprägter Forschungsorientierung;
- Entwicklung transdisziplinärer Dialogkompetenz;
- Aufbau von praxisorientierter Problemlösungskompetenz;
- Entwicklung von personalen und Sozialkompetenzen;
- Förderung der Bereitschaft zur Wahrnehmung gesellschaftlicher Verantwortung auf der Grundlage der erworbenen Kompetenzen.

Der Master-Studiengang Angewandte Informatik wird von der Fakultät für Mathematik und Informatik getragen. Der Master-Studiengang ist forschungsorientiert. Er vertieft und verbreitert die Fachkenntnisse, befähigt zum selbstständigen wissenschaftlichen Arbeiten, legt die Voraussetzungen zur Weiterentwicklung des Faches und bereitet auf eine anspruchsvolle Berufstätigkeit oder eine Promotion vor. AbsolventInnen sind qualifiziert für eigenverantwortliche und leitende Tätigkeiten. Der Master-Studiengang erlaubt eine sehr freie Gestaltung des Studiums um sowohl einen frühen Einstieg in forschungsnahen als auch innovative praktische Themengebiete zu ermöglichen. Weiterhin ermöglicht er eine Vertiefung in Themengebieten der Informatik, die insbesondere in den Natur-, Lebens- und Geisteswissenschaften zur Anwendung kommen.

Qualifikationsziele des Master-Studiengangs Angewandte Informatik

Die Absolventen des Studiengangs sollen nach Abschluss des Studiums folgende grundlegende Kompetenzen überfachlicher Art im Kontext der Informatik besitzen.

- Sie besitzen Problemlösungskompetenz und beherrschen die Wissensanwendung im Bereich der Informatik und zusätzlich in einem breiteren fachlichen Zusammenhang oder verwandten Disziplinen. Darüber hinaus sind sie in der Lage, diese Kompetenzen auch in neuen, unvertrauten Situationen anzuwenden.
- Sie haben die Kompetenz zur Arbeit in einem Team sowie zur Übernahme von herausgehobener Verantwortung in einem Team (Teamleitung).

- Sie können eigene Schlussfolgerungen auf aktuellem Stand von Forschung und Anwendung vermitteln und sich fachbezogen auf wissenschaftlichem Niveau austauschen.
- Sie besitzen die Kompetenz zu selbständiger Informationssammlung, Urteilsfähigkeit und selbständiger Aneignung von Wissen im Bereich der Informatik sowie verwandten Disziplinen. Insbesondere sind sie befähigt zur Rezeption und Interpretation von Forschungsliteratur und zur Bewertung alternativer Lösungsansätze in fachlicher Hinsicht im Bereich der Informatik als auch fachübergreifend.
- Darüber hinaus beherrschen sie den effektiven Umgang mit komplexen Fachproblemen und Situationen, verfügen über Entscheidungsfähigkeit, sowie können selbständig forschungs- oder anwendungsorientierte Projekte durchführen.
- Sie können in fachlichen Angelegenheiten mündlich und schriftlich in deutscher und englischer Sprache effektiv kommunizieren.

In fachlicher Hinsicht beherrschen die AbsolventInnen des Master-Studiengangs Angewandte Informatik insbesondere die Kompetenzen der Bachelor-AbsolventInnen, im Detail:

- Sie verfügen über Kenntnisse der Praktischen, Theoretischen, Technischen und Angewandten Informatik und der Methoden der Mathematik und können diese zur Lösung von konkreten informatischen Problemen anwenden.
- Sie können eine informatische Aufgabe eigenverantwortlich planen, durchführen, dokumentieren und präsentieren.
- Sie können innerhalb einer vorgegebenen Frist ein Problem aus dem Bereich der Informatik mit wissenschaftlichen Methoden bearbeiten und Lösungsvorschläge entwickeln und präsentieren.
- Sie beherrschen wissenschaftlich fundierte Methoden der Programmierung und können diese in Projekten praktisch anwenden. Dazu gehören die wissenschaftlichen Methoden des Entwurfs, der Implementierung und des Debuggens von Software.
- Sie kennen die Konzepte für den Entwurf und die Analyse von effizienten Algorithmen und können diese bei der Erstellung von Software selbständig einsetzen.
- Sie kennen die Grundlagen der Verwendung von Betriebssystemen und Verwaltung von Ressourcen und sind in der Lage, diese Kenntnisse bei dem Entwurf, der Umsetzung und der Optimierung von informatischen Systemen einzusetzen.
- Sie kennen die Probleme und Bedeutung der Verlässlichkeit in modernen Computersystemen und Rechenverbunden und können diese Kenntnisse bei der Planung, Umsetzung als auch der Pflege solcher Systeme praktisch berücksichtigen.

Zusätzlich beherrschen die AbsolventInnen des Master-Studiengangs Angewandte Informatik folgende fachliche Qualifikationen über die Lernergebnisse des Bachelorstudiengangs hinaus.

- Sie sind in der Lage, umfangreiche informatische Systeme unter vorgegebenen technischen und ökonomischen Randbedingungen selbständig zu planen, zu entwerfen und zu evaluieren, sowie dazugehörige Softwareprojekte zu leiten.
- Sie verfügen über vertiefte Kenntnisse in einem oder mehreren Spezialgebieten der Informatik wie Datenanalyse, Requirements Engineering, Verteilte Systeme, Informationssysteme, und können diese Kenntnisse bei dem Entwurf und der Entwicklung von informatischen Systemen praktisch einsetzen.
- Sie können komplexe informatische Systeme in abstrakte Komponenten (Software und Hardware) zerlegen und dafür Realisierungsmöglichkeiten gemäß vorgegeben Randbedingungen ermitteln und bewerten, sowie diese Realisierung planen und umsetzen.
- Sie sind in der Lage, sich selbständig in zukünftige Techniken der Informatik also auch fachübergreifende Gebiete einzuarbeiten, diese in Projekten anzuwenden, sie fachlich zu kommunizieren, und in wissenschaftlicher Hinsicht zu entwickeln.

Aktuelle Forschungsschwerpunkte und Details zum Master-Studium Angewandte Informatik finden sich auf der Webseite www.informatik.uni-heidelberg.de.

Einige Erläuterungen zu den Modulen

Begründung für Module mit weniger als 5 LP:

In diesem Studiengang gibt es einige Module mit weniger als 5 Leistungspunkten. Bei diesen Modulen handelt es sich um inhaltlich abgeschlossene Studieneinheiten, die nicht sinnvoll mit anderen Modulen zusammengelegt werden können.

Erläuterungen zur Sprache der Module:

Die Modulbeschreibungen sind in der Sprache verfasst in welcher das Modul auch angeboten wird, d.h. bei Modulbeschreibungen in Deutsch wird das Modul auch auf Deutsch gehalten, bei englischer Beschreibung auf Englisch. Ausnahmen sind möglich.

Beschreibung der Lehr- und Lernformen

Vorlesung: Präsentation des Lehrstoffs durch den Lehrenden mittels geeigneter Medien, Interaktion und Nachfragen möglich

Übung: Übungsaufgaben und kleinere Teile des Lehrstoffs werden erläutert, Nachfragen, Interaktion und Diskussion von und mit den Studierenden zum Verständnis des Lehrstoffs und der Beispielaufgaben

Seminar: Selbstständiges Erarbeiten eines wissenschaftlichen Themas, Erstellen einer Präsentation, Halten des Vortrags mit anschließenden Fragen und Diskussion der Teilnehmer zum Vortrag

Praktikum: Projektarbeit anhand einer Programmieraufgabe, selbstständiges Erstellen einer Software inklusive Dokumentation, Anfertigen eines Projektberichts und eines Vortrags, Halten des Vortrags zur Präsentation der Software

Inhaltsverzeichnis

1	Pflichtbereich	8
	Wissenschaftliches Arbeiten	9
	Seminar	11
	Masterarbeit	12
	Anwendungsgebiet	13
2	Wahlpflichtbereich	14
2.1	Gebietszuordnung der Module	14
2.2	Vertiefungen	17
	Bildverarbeitung	17
	Computergraphik und Visualisierung	18
	Information Systems Engineering	20
	Theoretische Informatik	22
	Wissenschaftliches Rechnen	23
2.3	Module aus der Informatik	24
	3D Computer Vision	25
	Advanced Machine Learning	27
	Algorithm Engineering	28
	Algorithmen für Geometrie und Topologie	30
	Algorithmische Geometrie	31
	Algorithms and Data Structures 2	32
	Artificial Intelligence	34
	Artificial Intelligence for Programming	35
	Big Data	37
	Cloud Computing 1	39
	Complex Network Analysis	41
	Computational Geometry	43
	Computer Graphics	44
	Computergraphik 1	46
	Computergraphik 2	48
	Computerspiele	49
	Deep Vision	50
	Discrete Structures 1	52
	Discrete Structures 2	54
	Formale Sprachen und Automatentheorie	55
	Fortgeschrittenenpraktikum	58
	Fundamentals of Machine Learning	59
	Geometric Modeling and Animation	60

Hardware Aware Scientific Computing	62
Inverse Problems	64
IT-Projektmanagement	65
IT-Sicherheit	67
Knowledge Discovery in Databases	69
Machine Learning	71
Mining Massive Datasets	73
Object-Oriented Programming for Scientific Computing	75
Object Recognition and Image Understanding	76
Optimization for Machine Learning	78
Praktische Geometrie	80
Projektseminar Biomedizinische Bildanalyse	82
Qualitätsmanagement	84
Randomisierte Algorithmen	85
Requirements Engineering	87
Scientific Visualization	88
Software Evolution	90
Software Ökonomie	92
Text Analytics	94
Verteilte Systeme I	96
Visualisierung im Bereich Cultural Heritage	98
Volumenvisualisierung	99
Wissensmanagement und Entscheidungen im Software Engineering	101
2.4 Module aus dem B.Sc./M.Sc. Mathematik	103
2.5 Module aus dem M.Sc. Technische Informatik	105

1 Pflichtbereich

Im Master-Studiengang Angewandte Informatik ist das Modul *Wissenschaftliches Arbeiten* ein Pflichtmodul. Weiterhin müssen auch zwei Seminare absolviert werden. Nachfolgend werden beide Module sowie die Module *Masterarbeit* und *Anwendungsgebiet* beschrieben.

Wissenschaftliches Arbeiten

Code IWA	Name Wissenschaftliches Arbeiten	
Leistungspunkte 2 LP	Dauer ein Semester	Turnus jedes Wintersemester
Lehrform 2 SWS	Arbeitsaufwand 60 h; davon 30 h Präsenzstudium 30 h Selbststudium und praktische Übungen (eventuell in Gruppen)	Verwendbarkeit M.Sc. Angewandte Informatik
Lernziele	<p>Die Studierenden</p> <p>kennen die wichtigsten Literaturquellen der Informatik; wissen, welche Tools und Techniken zur Verwaltung von Literatur existieren und wie diese verwendet werden; sind in der Lage, wissenschaftliche Texte (z.B. aus Tagungsbänden oder Journals) und Vorträge kritisch zu lesen und zu bewerten und sie kompakt zusammenzufassen; kennen die einschlägigen Techniken zur Präsentation eines wissenschaftlichen Vortrags; kennen die Möglichkeiten des wissenschaftlichen Publizierens und die Organisation von wissenschaftlichen Tagungen; kennen unterschiedliche Forschungsmethoden; kennen die Möglichkeiten einer Tätigkeit im wissenschaftlichen Umfeld nach dem Studium; kennen aktuelle Forschungsarbeiten in der Informatik; haben einen Überblick über die Wege der Finanzierung von Forschungsarbeiten; kennen die Anforderungen an die Struktur von Anträgen zur Forschungsförderung.</p>	
Inhalt	<p>Literaturrecherche und -verwaltung Wissenschaftliches Vortragen, Schreiben, Publizieren und Begutachten Forschungsförderung über Drittmittel Forschungsmethoden und aktuelle Forschungsprojekte Wissenschaftliches Arbeiten nach dem Studium</p>	
Voraussetzungen	keine	
Pruefungs- modalitaeten	Regelmäßige Teilnahme, Teilnahme an praktischen Übungen (u.a. durch Bearbeitung von Hausaufgaben und Projekten)	
Vergabe der LP	Bestehen der Modulprüfung	

Nuetzliche Literatur	
---------------------------------	--

Seminar

Code IS	Name Seminar	
Leistungspunkte 4 LP	Dauer ein Semester	Turnus jedes Semester
Lehrform Seminar 2 + 2 SWS (Seminar/ Tutorium)	Arbeitsaufwand 120 h; davon 30 h Präsenzstudium 90 h Vorbereitung Vortrag	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
Lernziele	<p>Kenntnis von Techniken des wissenschaftlichen Schreibens (insbesondere auch Literaturrecherche) Fähigkeit, komplexe wissenschaftliche Literatur zu erschließen</p> <p>Erweiterte Fähigkeit, komplexe wissenschaftliche Literatur in einem Vortrag zu präsentieren</p> <p>Erweiterte Fähigkeit, zu Vorträgen zu diskutieren und Feedback zu geben</p> <p>Fähigkeit, ein kurze wissenschaftliche Ausarbeitung zu einem komplexen Thema zu erstellen</p>	
Inhalt	<p>Einführung in und Einübung von Techniken des wissenschaftlichen Schreibens</p> <p>Vertiefte Einübung der Erschließung und Präsentation wissenschaftlicher Literatur</p> <p>Fortgeschritteneres Informatikthema</p>	
Voraussetzungen	empfohlen sind: Kenntnisse im Themengebiet des Seminars	
Prüfungsmodalitäten	Ausarbeitung und Halten eines Vortrages von etwa 60 Minuten Dauer (inklusive Diskussion), schriftliche Ausarbeitung von ca. 10 Seiten	
Vergabe der LP	Bestehen der Modulprüfung	
Nützliche Literatur		

Masterarbeit

Code IMa	Name Masterarbeit	
Leistungspunkte 30 LP	Dauer ein Semester	Turnus
Lehrform 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	Verwendbarkeit M.Sc. Angewandte Informatik
Lernziele	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	
Voraussetzungen	nach Prüfungsordnung mindestens 45 LP; weiterhin sind empfohlen: Wahlpflichtvorlesungen und Module Seminar (IS) und Fortgeschrittenenpraktikum (IFP)	
Pruefungs- modalitaeten	regelmäßige Treffen mit der/dem BetreuerIn, schriftliche Ausarbeitung und Präsentation	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur		

Anwendungsgebiet

Code IAG	Name Anwendungsgebiet	
Leistungspunkte 18 LP	Dauer	Turnus
Lehrform Vorlesung, Übung und / oder Praktikum	Arbeitsaufwand 540 h, Aufteilung in Präsenz- / Übungs- und Praktikumszeit in Absprache mit den Dozierenden	Verwendbarkeit M.Sc. Angewandte Informatik
Lernziele	Vertiefte Kenntnisse und Fähigkeiten in einem Anwendungsgebiet	
Inhalt	<p>Wahl eines Anwendungsgebietes nach den Regeln der Prüfungsordnung Festlegung von und Teilnahme an Modulen aus dem Anwendungsgebiet (die LP entsprechen dabei den 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. Festlegung eines Dozenten / einer Dozentin aus dem Anwendungsgebiet und der Informatik Gemeinsame Festlegung des Projektziels durch die Dozentinnen bzw. Dozenten und den/die Studierende. Das Projektziel umfasst eine informatische Leistung im Anwendungsgebiet.</p> <p>Festlegung des Arbeitsaufwandes und damit der LP für das Projekt Durchführung des Projekts entsprechend des Projektziels unter Anleitung der Dozentinnen bzw. Dozenten Dokumentation des Ergebnisses Erstellung eines Projektbericht Präsentation des Ergebnisses</p>	
Voraussetzungen	empfohlen ist das gleiche Anwendungsgebiet wie im Bachelor	
Prüfungsmodalitäten	<p>Die Prüfungsleistungen können durch nicht-informatische Module auf Bachelor- oder Masterniveau erbracht werden. Von den 18 LP können bis zu 10 LP durch ein interdisziplinäres Projekt erbracht werden.</p> <p>Prüfungsleistungen in dem Anwendungsgebiet und (optional) Prüfungsleistungen für das interdisziplinäre Projekt analog zum Modul IFP werden gewichtet nach dem jeweiligen Anteil der LP. Die Module im IAG müssen benotet sein, unbenotete Module werden nur in begründeten Ausnahmefällen zugelassen.</p>	
Vergabe der LP	Bestehen der Modulprüfung	
Nützliche Literatur		

2 Wahlpflichtbereich

Im Folgenden sind die Wahlpflichtmodule 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 Gebiete abzudecken. Die Zuordnung der Module zu den Gebieten ist im nachfolgenden Kapitel beschrieben. Anschließend folgen die Beschreibungen der Vertiefungen und dahinter die einzelnen Modulbeschreibungen.

2.1 Gebietszuordnung der Module

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

Bildverarbeitung BV)
Computergraphik und Visualisierung (CGV)
Datenbanksysteme (DB)
Optimierung
Parallele und Verteilte Systeme (PVS)
Software Engineering (SWE)
Technische Informatik (TI)
Theoretische Informatik (TH)
Wissenschaftliches Rechnen (WR)

Nachfolgend werden die Module den einzelnen Gebieten zugeordnet.
Alle nicht genannten Module sind keinem speziellen Gebiet zugeordnet.

Bildverarbeitung

3D Computer Vision (I3dCVi)
Advanced Machine Learning (IAML)
Artificial Intelligence (IAI)
Artificial Intelligence for Programming (IAIP)
Deep Vision (IDV)
Fundamentals of Machine Learning (IFML)
Machine Learning (IML)
Object Recognition and Image Understanding (IORIU)
Projektseminar Biomedizinische Bildanalyse (IPBB)

Computergraphik und Visualisierung (CGV)

Algorithmen für Geometrie und Topologie (IAGT)
Algorithmische Geometrie (IAGeo)
Artificial Intelligence (IAI)
Computational Geometry (ICGeo)
Computer Graphics (ICG)
Computergraphik 1 (ICG1)
Computergraphik 2 (ICG2)
Computerspiele (ICS)
Geometric Modeling and Animation (IGMA)
Inverse Probleme (IIP)
Praktische Geometrie (IPGeo)
Scientific Visualization (ISV)
Visualisierung im Bereich Cultural Heritage (IVCH)
Volume Visualization (IVV)

Datenbanksysteme (DB)

Complex Network Analysis (ICNA)
Knowledge Discovery in Databases (IKDD)
Text Analytics (ITA)

Optimierung

Lineare Optimierung (MD3)
Optimization for Machine Learning (IOML)

Parallele und Verteilte Systeme

Big Data (IBD)
Cloud Computing 1 (ICC1)
Mining Massive Datasets (IMMD)
Verteilte Systeme I (IVS1)

Software Engineering (SWE)

IT-Projektmanagement (IPM)
IT-Sicherheit (IITS)
Qualitätsmanagement (ISWQM)
Requirements Engineering (ISWRE)
Software Evolution (ISWEvolv)
Software Ökonomie (ISWÖk)
Wissensmanagement und Entscheidungen im Software Engineering (ISWKM)

Technische Informatik

Dieses Gebiet umfasst alle Grundlagen- und Vertiefungsmodule des Masters Technische Informatik.

Theoretische Informatik

Algorithm Engineering (IAE)
Algorithms and Data Structures 2 (IADS2)
Berechenbarkeit und Komplexität I (MM41)
Berechenbarkeit und Komplexität II (MM42)
Discrete Structures 1 (IDS1)
Discrete Structures 2 (IDS2)
Formale Sprachen und Automatentheorie (IFSA)
Mathematische Logik (MB9)
Randomisierte Algorithmen (IRA)

Wissenschaftliches Rechnen

Advanced Machine Learning (IAML)
Fundamentals of Machine Learning (IFML)
Hardware Aware Scientific Computing (IHASC)
Machine Learning (IML)
Numerik (MD1)
Object-Oriented Programming for Scientific Computing (IOPSC)

2.2 Vertiefungen

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. Ansprechpartner für diese Vertiefung sind Professor Dr. Hamprecht und Professor Dr. Ommer. Vor Beginn der Vertiefung wird ein Beratungsgespräch bei einem der Ansprechpartner empfohlen.

Die Veranstaltungen aus dem Gebiet der Bildverarbeitung sind:

- Artificial Intelligence (IAI) 6 LP
- Bildverarbeitung (MWInf6 oder MH19) 8 LP
- Object Recognition and Image Understanding (IORIU) 8 LP

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
2 Seminare - Bildverarbeitung	1 - 3	8
Bildverarbeitung (MWInf6 oder MH19)	1 - 3	8
Weitere Module aus dem Gebiet der Bildverarbeitung	1 - 3	28
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 sollten aus zwei ergänzenden Lehrgebieten gewählt werden.

Ansprechpartner für diese Vertiefung sind Herr Professor Dr. Sadlo und Frau Dr. Krömker. Vor Beginn der Vertiefung wird ein Beratungsgespräch bei einem der Ansprechpartner empfohlen.

Vertiefende Module aus dem Gebiet CGV sind:

- Algorithmen für Geometrie und Topologie (IAGT) 4 LP
- Algorithmische Geometrie (IAGeo) 4 LP
- Computational Geometry (ICGeo) 8 LP
- Computer Graphics (ICG) 8 LP
- Computergraphik 1 (ICG1) 6 LP
- Computergraphik 2 (ICG2) 6 LP
- Computerspiele (ICS) 8 LP
- Geometric Modeling and Animation (IGMA) 8 LP
- Praktische Geometrie (IPGeo) 4 LP
- Scientific Visualization (ISV) 8 LP
- Visualisierung im Bereich Cultural Heritage (IVCH) 2 LP
- Volume Visualization (IVV) 8 LP

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
Wissenschaftliches 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 anderen Gebieten	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

Diese Vertiefung befähigt zu Entwicklung, Betrieb und Wartung von komplexen Informationssystemen. Sie kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst die Seminare, Praktika, Bachelor- bzw. Masterarbeiten und die Wahlpflichtveranstaltungen. Dabei sind die Module zu gleichen Teilen auf die Lehr- und Forschungsgebiete der beiden Arbeitsgruppen Datenmanagement und -analyse (DMA) 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. Dazu gehören u.a. Konzepte und Methoden aus den Bereichen Daten- und Textanalyse, Informationsnetzwerke, Datenmanagement, Softwarequalität und Requirements Engineering. Das ISE-Projekt wird immer im Wintersemester angeboten. Ansprechpartner für diese Vertiefung sind Herr Professor Dr. Gertz (DMA) und Frau Professor Dr. Paech (SWE). Vor Beginn der Vertiefung wird ein Beratungsgespräch bei einem der Ansprechpartner empfohlen.

Vertiefende Module aus dem Gebiet SWE:

- Requirements Engineering (ISWRE) 8 LP
- Qualitätsmanagement (ISWQM) 8 LP
- IT-Projektmanagement (IPM) 3 LP
- Software-Ökonomie (ISWÖk) 3 LP
- Software-Evolution (ISWEvolv) 3 LP
- Wissensmanagement in der Softwareentwicklung (ISWKM) 3 LP

Vertiefende Module aus dem Gebiet DMA:

- Complex Network Analysis (ICNA) 8 LP
- Knowledge Discovery in Databases (IKDD) 8 LP
- Text Analytics (ITA) 8 LP

Ergänzende Module:

- Artificial Intelligence (IAI) 6 LP
- Big Data (IBD) 3 LP
- Cloud Computing 1 (ICC) 3 LP
- Fundamentals of Machine Learning (IFML) 8 LP
- IT-Sicherheit (IITS) 8 LP
- Machine Learning (IML) 8 LP
- Mining Massive Datasets (IMMD) 6 LP
- Object-Oriented Programming for Scientific Computing (IOPSC) 6 LP
- Scientific Visualization (ISV) 8 LP
- Verteilte Systeme I (IVS1) 6 LP

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
Seminar - Lehrgebiet SWE	1 - 3	4
Seminar - Lehrgebiet DMA	1 - 3	4
ISE-Projekt	1 - 3	16
2 Vertiefende Module Lehrgebiet SWE	1 - 3	16
2 Vertiefende Module Lehrgebiet DMA	1 - 3	16
Weitere Module aus SWE oder DMA	1 - 3	6
Ergänzende Module 1	1 - 3	8
Masterarbeit SWE oder DMA	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 DMA und SWE gemacht werden. 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 4 vertiefende Vorlesungen aus dem Gebiet der Theoretischen Informatik zu wählen und die Masterarbeit in diesem Gebiet anzufertigen. Weitere Lehrveranstaltungen in der Informatik sowie das Anwendungsgebiet können gemäß der Prüfungsordnung frei gewählt werden. Ansprechpartner für diese Vertiefung sind Privatdozent Dr. Merkle und Juniorprofessor Joos. Vor Beginn der Vertiefung wird ein Beratungsgespräch empfohlen.

Vertiefende Vorlesungen aus dem Bereich Theoretische Informatik:

- Algorithm Engineering (IAE) 8 LP
- Algorithms and Data Structures 2 (IADS2) 8 LP
- Berechenbarkeit und Komplexität 1 (MM41) 8 LP
- Berechenbarkeit und Komplexität 2 (MM42) 8 LP
- Discrete Structures 1 (IDS1) 8 LP
- Discrete Structures 2 (IDS2) 8 LP
- Formale Sprachen und Automatentheorie (IFSA) 8 LP

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
2 Seminare - Lehrgebiet Theoretische Informatik	1 - 3	8
4 Vertiefende Vorlesungen - Lehrgebiet Theoretische Informatik	1 - 3	30
Weitere Module Informatik (Wahlpflichtmodule)	1 - 3	32
Masterarbeit - Lehrgebiet Theoretische Informatik	4	30
Anwendungsgebiet	1 - 3	18
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. Ansprechpartner für diese Vertiefung ist Herr Professor Dr. Bastian. Vor Beginn der Vertiefung wird ein Beratungsgespräch empfohlen.

Vertiefende Vorlesungen aus dem Gebiet Wissenschaftliches Rechnen:

- Object-Oriented Programming for Scientific Computing (IOPSC) 6 LP
- Numerik 1 (MD1) 8 LP

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

Hier können auch einige der Module aus dem Grundmodul, dem Aufbaumodul und dem Spezialisierungsmodul Numerik und Optimierung gehört werden.

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
2 Seminare Wissenschaftliches Rechnen	1 - 3	8
Object-Oriented Programming for Scientific Computing (IOPSC)	1 - 3	6
Weitere Module aus dieser Vertiefung	1 - 3	40
Weitere Module aus mindestens 1 anderem Gebiet (empfohlen: SWE)	1 - 3	16
Masterarbeit Wissenschaftliches Rechnen	4	30
Anwendungsgebiet	1 - 3	18
LP Summe		120

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

2.3 Module aus der Informatik

Nachfolgend sind die Module aus der Informatik in alphabetischer Reihenfolge beschrieben.

3D Computer Vision

Code I3DCVi	Name 3D Computer Vision	
Leistungspunkte 6 LP	Dauer one semester	Turnus every winter semester
Lehrform Lecture 2 SWS, Exercise 2 SWS	Arbeitsaufwand 180 h; thereof 30 h lectures 30 h exercises 20 h revision and home exercise 70 h programming a mini research project 30 h preparation of final report	Verwendbarkeit B.Sc. Angewandte Informatik M.Sc. Angewandte Informatik M.Sc. Scientific Computing
Lernziele	<p>The students</p> <ul style="list-style-type: none"> - Understand the principles behind estimating 3D Point Clouds and Motion from two or more images. They are able to apply this knowledge to new tasks in the field of 3D reconstruction. - Understanding the principles of an image processing, the image formation process and corresponding Geometry. This can be utilized to design new algorithms, for e.g. 3D motion estimation for autonomous driving. - Understand and implement methods that combine machine learning-based methods with classical computer vision-based techniques. - Have studied various state-of-the-art computer vision systems and approaches, and are then able to evaluate and classify new systems and approaches. - Understand and implement different approaches for object tracking and object-instance recognition. 	
Inhalt	<p>This lecture covers areas of computer vision which deal with 3D reconstruction and scene understanding. This means, for instance, to recover a 3D scene from a set of photographs or video, or to extract and track objects in the scene. We discuss the underlying principles and methods to solve such tasks. In particular, we cover techniques from deep learning, traditional approaches, and mixtures of the two. We also introduce the necessary background knowledge, e.g. camera models, deep learning, image formation model, Kalman Filters, etc.</p>	
Voraussetzungen	<p>recommended is a basic machine learning background (e.g. Fundamentals of Machine Learning, Advanced Machine Learning or equivalent)</p>	

Pruefungs- modalitaeten	graded final report (about 10 pages) or final report (about 5 pages) together with oral exam
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	

Advanced Machine Learning

Code IAML	Name Advanced Machine Learning	
Leistungspunkte 8 LP	Dauer one semester	Turnus follows *Fundamentals of Machine Learning*
Lehrform 4 SWS lecture (in English), 2 SWS tutorial, homework assignments	Arbeitsaufwand 240h; thereof 90h lectures and tutorials 90h lecture wrap-up and homework 60h graded final report	Verwendbarkeit cannot be combined with *Machine Learning* B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing,
Lernziele	Students get to know advanced machine learning methods that define the state-of-the-art and major research directions in the field. Students understand when these methods are called for, what limitations of standard solutions they address, and how they are applied to real-world problems. In addition, students learn how to use Python-based machine learning software such as scikit-learn, theano and OpenGM.	
Inhalt	The lecture, along with its sibling *Fundamentals of Machine Learning*, offers an extended version of the one-semester course *Machine Learning*: Multi-layered architectures (neural networks, deep learning); directed and undirected probabilistic graphical models (Gaussian processes, latent variable models, Markov random fields, structured learning); feature optimization (feature selection and learning, dictionary learning, kernel approximation, randomization); weak supervision (one-class learning, multiple instance learning, active learning, reinforcement learning)	
Voraussetzungen	recommended are: lecture *Fundamentals of Machine Learning* or similar	
Pruefungsmodalitaeten	cannot be combined with *Machine Learning*, written exam (report on a 60h mini-research project)	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur	David Barber: Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012 Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, 2006	

Algorithm Engineering

Code IAE	Name Algorithm Engineering	
Leistungspunkte 8 LP	Dauer one semester	Turnus every summer semester
Lehrform Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand 240h; thereof 90h lectures and tutorials, 15h exam preparations, 135h lecture wrap-up and homework	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	<p>Students obtain a systematic understanding of algorithmic questions and solution approaches in the area of algorithm engineering.</p> <p>The students will be able to transfer the learned techniques onto similar problems and be able to interpret and understand current research topics in the area of algorithm engineering.</p> <p>Given a real-world problem, students are able to select appropriate algorithms to come up with and implement efficient solutions.</p> <p>In particular, students know realistic machine models and applications, algorithm design, implementation techniques, experimental methodology and can interpret of measurements.</p>	
Inhalt	<p>The listed abilities will be learned by concrete examples. In particular, we will almost always cover the best practical and theoretical methods. This methods often deviate a lot by the algorithms learned in the basic courses. To this end the lecture covers FPT/Kernelization in practice (independent set, vertex cover, (all) minimum cuts (NOI algorithm), clique cover, node ordering), multi-level algorithms (graph partitioning, modularity clustering, dynamic clustering, process mapping, spectral techniques, exact approaches), route planning (contraction hierarchies, arc-flags, hub-label algorithm), dynamic graph algorithms (single-source reachability, transitive closure, matching, minimum cuts, graph generation).</p>	
Voraussetzungen	<p>recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Mathematik für Informatiker 1 oder Lineare Algebra 1 (MA4), Algorithms and Data Structures 2</p>	
Prüfungsmodalitäten	<p>Successful participation in the exercises (at least 50% of total achievable points) and passing an oral exam</p>	

Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	<p>Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms, 3rd Edition. MIT Press 2009, ISBN 978-0-262-03384-8, pp. I-XIX, 1-1292</p> <p>Jon M. Kleinberg, Éva Tardos: Algorithm design. Addison-Wesley 2006, ISBN 978-0-321-37291-8, pp. I-XXIII, 1-838</p> <p>Stefan Näher: LEDA, a Platform for Combinatorial and Geometric Computing. Handbook of Data Structures and Applications 2004</p>

Algorithmen für Geometrie und Topologie

Code IAGT	Name Algorithmen für Geometrie und Topologie	
Leistungspunkte 4 LP	Dauer ein Semester	Turnus unregelmäßig
Lehrform Vorlesung 2 SWS, 1 SWS Übung	Arbeitsaufwand 120 h; davon 45 h Präsenzstudium 15 h Prüfungsvorbereitung 60 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	Verwendbarkeit kann nicht kombiniert werden mit *Computational Geometry* (ICGeo) B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	Die Studierenden kennen die grundlegenden Algorithmen der geometrischen und topologischen Datenverarbeitung. Sie verstehen Grundkonzepte wie das Bestimmen von Isoflächen, simpliziale Komplexe und Bettizahlen, und können sie algorithmisch umzusetzen. Sie beherrschen die zugehörigen Datenstrukturen und können die Komplexität der verschiedenen Algorithmen berechnen.	
Inhalt	Basiskonzepte aus Geometrie und Topologie, Sichtbarkeitsanalysen, Alpha-Shape, Isoflächen, Diskrete Mannigfaltigkeiten und Morse Theorie	
Voraussetzungen	empfohlen ist: Algorithmen und Datenstrukturen (IAD)	
Prüfungsmodalitäten	eine schriftliche oder mündliche Abschlussprüfung	
Vergabe der LP	Bestehen der Modulprüfung	
Nützliche Literatur	Rolf Klein: Algorithmische Geometrie, Springer Verlag, 2005. Herbert 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	

Algorithmische Geometrie

Code IAGeo	Name Algorithmische Geometrie	
Leistungspunkte 4 LP	Dauer ein Semester	Turnus unregelmäßig
Lehrform Vorlesung 2 SWS, 1 SWS Übung	Arbeitsaufwand 120 h; davon 45 h Präsenzstudium 15 h Prüfungsvorbereitung 60 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	Verwendbarkeit kann nicht kombiniert werden mit *Computational Geometry* (ICGeo) B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik M.Sc. Scientific Computing
Lernziele	Die Studierenden kennen die grundlegenden Algorithmen und Datenstrukturen der geometrischen Datenverarbeitung. Sie verstehen Grundkonzepte wie konvexe Hülle, Schnittpunktbestimmung und effektive Punktsuche und können sie algorithmisch umzusetzen. Sie beherrschen die wesentlichen Datenstrukturen zur effizienten Speicherung und Weiterverarbeitung und können die Komplexität der verschiedenen Algorithmen berechnen.	
Inhalt	Basiskonzepte Effiziente Punktsuche, Sweep-Algorithmen, Voronoidiagramme Delaunaytriangulierung Allgemeine Suchstrukturen	
Voraussetzungen	empfohlen ist: Algorithmen und Datenstrukturen (IAD)	
Pruefungs- modalitaeten	eine schriftliche oder mündliche Abschlussprüfung	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur	Rolf Klein: Algorithmische Geometrie, Springer Verlag, 2005. Herbert 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	

Algorithms and Data Structures 2

Code IADS2	Name Algorithms and Data Structures 2	
Leistungspunkte 8 LP	Dauer one semester	Turnus every winter semester
Lehrform Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand 240h; thereof 90h lectures and tutorials, 15h exam preparations, 135h lecture wrap-up and homework	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	<p>Students:</p> <ul style="list-style-type: none"> - understand fundamental theoretical and practical concepts of advanced algorithms and data structures, - get to know established methods and algorithms, - are familiar with issues of efficient implementations, - are able to identify/formulate algorithmic problems in/for different application areas, - are able to analyse new algorithms as well as analysing their running time, and select appropriate algorithms for applications - are able to apply algorithms and data structures to real-world problems, and can objectively assess the quality of the results 	
Inhalt	<p>Introduction to Algorithm Engineering:</p> <ul style="list-style-type: none"> - advanced data structures (efficient addressable priority queues, monotone priority queues, external priority queues), - advances graph algorithms (strongly connected components, shortest paths, maximum flows / min s-t cuts, min-cost flows), techniques to solve problems to optimality (branch-and-bound, branch-and-reduce, dynamic programming, integer linear programming as a modelling tool), - introduction to randomized algorithms, greedy algorithms, approximation algorithms, advanced string algorithms, geometric algorithms, external memory algorithms 	
Voraussetzungen	<p>recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Mathematik für Informatiker 1 (IMI1) oder Lineare Algebra 1 (MA4)</p>	
Prüfungsmodalitäten	<p>Successful participation in the exercises (at least 50% of total achievable points) and passing a written exam</p>	
Vergabe der LP	<p>Bestehen der Modulprüfung</p>	

Nuetzliche Literatur	<p>Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms, 3rd Edition. MIT Press 2009, ISBN 978-0-262-03384-8, pp. I-XIX, 1-1292</p> <p>Kurt Mehlhorn, Peter Sanders: Algorithms and Data Structures: The Basic Toolbox. Springer 2008, ISBN 978-3-540-77977-3</p> <p>Jon M. Kleinberg, Éva Tardos: Algorithm design. Addison-Wesley 2006, ISBN 978-0-321-37291-8, pp. I-XXIII, 1-838</p> <p>Stefan Näher: LEDA, a Platform for Combinatorial and Geometric Computing. Handbook of Data Structures and Applications 2004</p>
---------------------------------	---

Artificial Intelligence

Code IAI	Name Artificial Intelligence	
Leistungspunkte 6 LP	Dauer one semester	Turnus irregular
Lehrform 2 SWS lecture, 2 SWS tutorial	Arbeitsaufwand 180 h; thereof 60h lectures and tutorials 50h graded final report of mini project 70h lecture wrap-up, programming exercises and homework	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	<p>To be able to develop algorithms for solving problems by informed/uninformed search.</p> <p>To know how to apply logical inference for finding solutions.</p> <p>To have a firm command of applying inference under uncertainty.</p> <p>To have gained a solid understanding of learning agents based on deep learning and neural networks.</p> <p>To know how to apply the developed expertise to different application areas such as Computer Vision or text mining.</p>	
Inhalt	<p>Solving problems by search</p> <p>Game playing</p> <p>Inference using logic</p> <p>Knowledge bases</p> <p>Planning and acting</p> <p>Reasoning under uncertainty</p> <p>Deep Learning and Neural Networks</p> <p>Machine Learning Applications</p>	
Voraussetzungen	recommended are: basic programming skills and knowledge of basic calculus, statistics, and linear algebra	
Pruefungsmodalitaeten	exam (written report on mini research project)	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur	Russell & Norvig: Artificial Intelligence: A Modern Approach	

Artificial Intelligence for Programming

Code IAIP	Name Artificial Intelligence for Programming	
Leistungspunkte 6 LP	Dauer ein Semester	Turnus at least every 4th semester
Lehrform Lecture 2 h + Exercise course 2 h	Arbeitsaufwand 180 h; thereof 60 h lecture 15 h preparation for exam 105 h self-study and working on assignments (optionally in groups)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	<p>Expected learning outcomes are:</p> <ul style="list-style-type: none"> - Knowledge of selected classical methods in artificial intelligence, in particular knowledge representation, search methods, rule systems - Basic knowledge about probabilistic models and probabilistic programming - Knowledge of techniques for code representation and parsing - Knowledge of techniques for modeling of code via neural networks - Knowledge of basic and advanced methods for program synthesis - Familiarity with semantic parsing and code summarization - Familiarity with selected applications of AI for programming, e.g. code-to-code translation, code recommendations, and detection of bugs in code. 	
Inhalt	<p>This module covers the following topics:</p> <ul style="list-style-type: none"> - introduction to classical methods in artificial intelligence, in particular knowledge representation, search methods, rule systems - introduction to probabilistic models and probabilistic programming - fundamentals of code representation and parsing - modeling of code via neural networks and sequence models/transformers - basic and advanced methods for program synthesis - introduction to semantic parsing and code summarization - state-of-the-art applications of AI for programming, e.g. code-to-code translation, code recommendations, detection of vulnerabilities in code. 	

Voraussetzungen	Skills in programming (preferably Python) and in elementary knowledge of probability theory / statistics. Recommended prerequisites are lectures in machine learning, e.g. Foundations of machine learning.
Pruefungsmodalitaeten	Successful participation in the exercises with home-work (achieving a minimum score of at least 50% of points) and passing a final exam.
Vergabe der LP	Pass the final exam.
Nuetzliche Literatur	Stuart J. Russell: Artificial intelligence: a modern approach, (3rd ed.), Pearson, 2016, Heidi: https://bit.ly/2V9LQT9 Noah D. Goodman, Joshua B. Tenenbaum: Probabilistic Models of Cognition (2nd ed.), 2016. Online: https://probmods.org/ Jeremy Howard: Deep learning for coders with fastai and PyTorch, (1st ed.), O'Reilly, 2020, Online via Heidi: https://bit.ly/3jUMkH7 Aurélien Géron: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, (2nd ed.), O'Reilly, 2019, Online via Heidi: https://bit.ly/3dVhieA

Big Data

Code IBD	Name Big Data	
Leistungspunkte 3 LP	Dauer ein Semester	Turnus unregelmäßig
Lehrform Vorlesung 2 SWS	Arbeitsaufwand 90 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Hausaufgabenbearbeitung	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
Lernziele	<p>Die Studierenden</p> <p>kennen den prinzipiellen Aufbau einer Big Data Anwendung haben die erforderlichen Grundlagen gelernt (Technologien, Methoden und Konzepte)</p> <p>sind mit den Sicherheitsproblematiken von Big Data Anwendungen vertraut und können eine adäquate Architektur konzipieren</p> <p>können Big Data Konzepte in Anwendungen umsetzen sind in der Lage, öffentliche und private Daten in Anwendungen zu aggregieren</p> <p>können die erworbenen Kenntnisse auf verschiedene Anwendungsbereiche übertragen</p>	
Inhalt	<p>Die Vorlesung bietet eine praktische Einführung in das aktuelle Gebiet Big Data, wobei auch Cloud Ressourcen im Internet für eigene Anwendungen transparent genutzt werden können. Schwerpunkte sind u.a.:</p> <p>Fundamentals: Volume, Variety, Velocity, Veracity, Visualization, Value</p> <p>Architecture: Hadoop as a Service, Data Warehouse</p> <p>Data Management: Movement, Monitoring, Provenance, Preservation, SLAs</p> <p>Data Publication: Preparation, Curation, Discovery, Open Data, Open Access</p> <p>Data Security, Privacy & Trust: Risk Management, Usage Control, Trusted Datastore</p>	
Voraussetzungen	empfohlen sind: Kenntnisse und Fähigkeiten wie sie im Modul ISW, IBN und ICC vermittelt werden. Modul IPDPA wird als Ergänzung bzw. Erweiterung empfohlen.	
Prüfungs- modalitäten	eine Klausur oder mündlichen Abschlussprüfung	

Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	BITKOM Leitfaden Big-Data-Technologien, 2014 (online) Bücher aus dem O'Reilly Data Science Starter Kit, 2014 (online) L.Barroso, J.Clidas, U.Hölzle, The Data Center as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Morgan&Claypool, 2013 (online) A.Murthy, V.Vavilapalli, D.Eadline, J.Niemiec, Apache Hadoop YARN: Moving beyond MapReduce and Batch Processing with Apache Hadoop 2, Addison Wesley Data&Analytics, 2014 C.Baun, M.Kunze, J.Nimis, S.Tai, Cloud Computing - Webbasierte dynamische IT-Services, Informatik im Fokus, Springer Verlag 2011, 2. Auflage C.Bengel, C.Baun, M.Kunze, U.Stucky, Masterkurs Verteilte und Parallele Systeme, Springer Verlag 2008

Cloud Computing 1

Code ICC1	Name Cloud Computing 1	
Leistungspunkte 3 LP	Dauer ein Semester	Turnus unregelmäßig
Lehrform Vorlesung 2 SWS	Arbeitsaufwand 90 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Hausaufgabenbearbeitung	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
Lernziele	<p>Die Studierenden</p> <p>kennen den prinzipiellen Aufbau einer Computing Cloud</p> <p>haben die erforderlichen Grundlagen gelernt (Virtualisierung, Web Services)</p> <p>sind mit den Sicherheitsproblematiken von Cloud-Systemen vertraut und können eine adäquate Architektur konzipieren</p> <p>können Konzepte wie IaaS, PaaS und SaaS in Anwendungen umsetzen</p> <p>sind in der Lage, öffentliche und private Cloud-Systeme zu nutzen</p> <p>können die erworbenen Kenntnisse auf verschiedene Anwendungsbereiche zu übertragen</p>	
Inhalt	<p>Die Vorlesung bietet eine praktische Einführung in das aktuelle Gebiet Cloud Computing, bei dem Ressourcen im Internet für eigene Anwendungen transparent genutzt werden können.</p> <p>Schwerpunkte sind u.a.:</p> <ul style="list-style-type: none"> Grundlagen Virtualisierung und Web Services Cloud Angebote im Internet (Taxonomie) Cloud Security Programmiermodelle Wiss. Rechnen in der Cloud High Performance Computing as a Service Big Data 	
Voraussetzungen	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in den Modulen Einführung in Software Engineering (ISW) sowie Betriebssysteme und Netzwerke (IBN) vermittelt werden	
Prüfungsmodalitäten	mündliche Abschlussprüfung	
Vergabe der LP	Bestehen der Modulprüfung	

Nuetzliche Literatur	C.Baun, M.Kunze, J.Nimis, S.Tai, Cloud Computing - Web basierte dynamische IT-Services, Informatik im Fokus, Springer Verlag 2011, 2. Auflage C.Bengel, C.Baun, M.Kunze, U.Stucky, Masterkurs Verteilte und Parallele Systeme, Springer Verlag 2008
---------------------------------	---

Complex Network Analysis

Code ICNA	Name Complex Network Analysis	
Leistungspunkte 8 LP	Dauer one semester	Turnus every 2nd wintersemester
Lehrform Lecture 4 SWS, Exercise course 2 SWS	Arbeitsaufwand 240 h; thereof 90 h lecture 12 h preparation for exam 130 h self-study and working on assignments/projects (optionally in groups)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing B.Sc. Mathematik
Lernziele	<p>Students</p> <ul style="list-style-type: none"> - can describe basic measures and characteristics of complex networks - can implement and apply basic network analysis algorithms using programming environments such as R or Python - can describe different network models and can describe, compute, and analyze characteristic parameters of these models - know how to compute different complex network measures and how to interpret these measures - know different generative models for constructing complex networks, especially scale-free networks - know the fundamental methods for the detection of communities in networks and the analysis of their evolution over time - are familiar with basic concepts of network robustness - understand the principles behind the spread of phenomena in complex networks 	

Inhalt	<ul style="list-style-type: none"> - Graph theory and graph algorithms; basic network measures - Random networks and their characteristics (degree distribution, component sizes, clustering coefficient, network evolution), small world phenomena - Scale-free property of networks, power-laws, hubs, universality - Barabasi-Albert model, growth and preferential attachment, degree dynamics, diameter and clustering coefficient - Evolving networks, Bianconi-Barabasi model, fitness, Bose-Einstein condensation - Degree correlation, assortativity, degree correlations, structural cutoffs - Network robustness, percolation theory, attack tolerance, cascading failures - Communities, modularity, community detection and evolution - Spreading phenomena, epidemic modeling, contact networks, immunization, epidemic prediction
Voraussetzungen	recommended are: Algorithmen und Datenstrukturen (IAD), Knowledge Discovery in Databases (IKDD), Lineare Algebra I (MA4)
Pruefungsmodalitaeten	final written exam
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	<ul style="list-style-type: none"> - Albert-Laszlo Barabasi: Network Science, Cambridge University Press, 2016. - M.E.J. Newmann: Networks: An Introduction, Oxford University Press, 2010. - Vito Latora, Vincenzo Nicosia, Giovanni Russo: Complex Networks - Principles, Methods and Applications, Cambridge University Press, 2017. - David Easley, Jon Kleinberg: Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010. - Stanley Wasserman, Katherine Faust: Social Network Analysis-Methods and Applications, Cambridge University Press, 1994.

Computational Geometry

Code ICGeo	Name Computational Geometry	
Leistungspunkte 8 LP	Dauer one semester	Turnus irregular
Lehrform Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand 240 h; thereof 90 h lectures and tutorials 15 h preparation for exam 135 h self-study and working on assignments/projects (optionally in Gruppen)	Verwendbarkeit cannot be combined with *Algorithmische Geometrie* (IAGeo) and *Algorithmen für Geometrie und Topologie* (IAGT) B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	The students know the basic algorithms and data structures of geometric and topological data processing. They understand basic concepts such as convex hull, determination of intersections, effective point search, the determination of isosurfaces, simplicial complexes and Betti numbers, and can implement them algorithmically. They master the associated data structures for efficient storage and further processing and can calculate the complexity of the various algorithms.	
Inhalt	Basic concepts from geometry and topology, efficient point search, general search structures, sweep algorithms, Voronoi diagrams, Delaunay triangulation, visibility analyzes, alpha-shape, isosurfaces, discrete manifolds and Morse theory	
Voraussetzungen	recommended is: Algorithmen und Datenstrukturen (IAD)	
Pruefungs- modalitaeten	written or oral exam at the end of the semester	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur	Rolf Klein: Algorithmische Geometrie, Springer Verlag, 2005 Herbert 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 current publications	

Computer Graphics

Code ICG	Name Computer Graphics	
Leistungspunkte 8 LP	Dauer one semester	Turnus every 3rd semester
Lehrform lecture 4 SWS, exercise 2 SWS	Arbeitsaufwand 240 h; thereof 90 h on-campus program 15 h exam preparation 135 h independent study and exercises (possibly in groups)	Verwendbarkeit cannot be combined with Computergraphik 1 and 2 (ICG1, ICG2) B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	The students understand fundamental and advanced concepts of computer graphics. They understand the mathematical fundamentals, data structures, and implementation aspects. They get to know raster graphics, geometric transforms, color perception and color models, and basics of geometric modeling. The students are able to apply these concepts to real-world problems using existing software packages, and develop small programs using OpenGL 4.	
Inhalt	<ul style="list-style-type: none"> - Introduction - Perception and Color - Raytracing - Transformations - Rasterization - OpenGL - Textures - Curves - Spatial Data Structures 	
Voraussetzungen	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
Prüfungsmodalitäten	cannot be combined with Computergraphik 1 and 2 (ICG1, ICG2), an oral or written exam	
Vergabe der LP	Bestehen der Modulprüfung	

Nuetzliche Literatur	<p>P. Shirley, S. Marschner: Fundamentals of Computer Graphics, 3rd Edition, AK Peters OpenGL Specifications(GL 4.5 + GLSL 4.50) http://www.opengl.org/registry/ Optional A. S. Glassner: An Introduction to Ray Tracing, Academic Press T. Akenine-Möller, E. Haines: Real-Time Rendering, AK Peters, 2008</p>
---------------------------------	---

Computergraphik 1

Code ICG1	Name Computergraphik 1	
Leistungspunkte 6 LP	Dauer ein Semester	Turnus jedes 2. Wintersemester
Lehrform Vorlesung 2 SWS, Übung 2 SWS	Arbeitsaufwand 180 h; davon 60 h Präsenzstudium, 100 h Aufgabenbearbeitung, 20 h Prüfungsvorbereitung	Verwendbarkeit kann nicht kombiniert werden mit Computer Graphics (ICG) B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
Lernziele	<p>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 können die physikalischen Grundlagen für globale Lichtmodelle in Fremdsoftware abrufen. Das sichere Beherrschen der Programmierung kleinerer Graphikprogramme und die Generierung von Einzelbildern, 3D Grafiken und Animationen für die unterschiedlichen Zwecke des wissenschaftlichen Rechnens wird auf Basis der Programmbibliothek OpenGL (Open Graphics Library) vermittelt.</p>	
Inhalt	<p>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</p>	

Voraussetzungen	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK)
Pruefungs-modalitaeten	kann nicht kombiniert werden mit Computer Graphics (ICG), eine mündliche oder schriftliche Abschlussprüfung
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche 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

Computergraphik 2

Code ICG2	Name Computergraphik 2	
Leistungspunkte 6 LP	Dauer ein Semester	Turnus jedes 2. Sommersemester
Lehrform Vorlesung 2 SWS, Übung 2 SWS	Arbeitsaufwand 180 h; davon 60 h Präsenzstudium, 100 h Aufgabenbearbeitung, 20 h Prüfungsvorbereitung	Verwendbarkeit kann nicht kombiniert werden mit Computer Graphics (ICG) B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
Lernziele	<p>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.</p> <p>Das sichere Beherrschen der Programmierung von Grafikkarten (Graphik-Hardware GPU) mit der Programmiersprache *C for graphics* und anderen *Application Programming Interfaces* (API) für globales Rendering wird unter Verwendung von numerischen Algorithmen zur Lösung nichtlinearer Gleichungssysteme vermittelt.</p>	
Inhalt	Shading und Bidirectional Reflectance Distribution Function (BRDF), C for graphics (Cg), Radiosity, Photon Mapping, Nichtphotorealismus (NPR), Splines	
Voraussetzungen	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Computergraphik 1 (ICG1)	
Pruefungs- modalitaeten	kann nicht kombiniert werden mit Computer Graphics (ICG), eine schriftliche Abschlussprüfung	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur	<p>J. D. Foley, A. Van Dam, S. K. Feiner, J . F. Hughes und R. L. Phillips: Computer Graphics: Principles and Practice. Addison-Wesley, 1996</p> <p>Randima Fernando, Mark J. Kilgard: The Cg Tutorial. Addison-Wesley, 2003</p>	

Computerspiele

Code ICS	Name Computerspiele	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus jedes Sommersemester
Lehrform Vorlesung 3 SWS, Übung 3 SWS	Arbeitsaufwand 240 h; davon 75 h Präsenzstudium 15 h Prüfungsvorbereitung 150 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	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 Vorstellung von OGRE als einer open-source Game Engine Graphik und Computerspiele: ein Überblick Kollisionserkennungstechniken Animationstechniken und Physik bei Computerspielen mit Fokus auf der open source Bibliothek Bullet Pfadplanung und KI	
Voraussetzungen	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
Pruefungs- modalitaeten	eine schriftliche oder mündliche Abschlussprüfung	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur	Gregory et al: Game Engine Architecture Ericson: Real-Time Collision Detection Eberly: Game Physics Millington: Artificial Intelligence for Games	

Deep Vision

Code IDV	Name Deep Vision	
Leistungspunkte 6 LP	Dauer one semester	Turnus irregular
Lehrform lecture 2 SWS, tutorial 2 SWS	Arbeitsaufwand 180h; thereof 60h lectures and tutorials 70h lecture wrap-up, programming excercises, and homework 50h preparation for project work, examination	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	<p>To have reached an understanding of the foundations underlying deep learning</p> <p>To be able to practically apply deep learning algorithms to new problems in Computer Vision</p> <p>To have a firm command of the algorithmic basics and to be able to analyze and solve new problems in novel application areas</p> <p>To be capable of understanding and analyzing the latest publications in deep learning and Computer Vision and to evaluate their strengths and weaknesses.</p> <p>To have reached understanding of state-of-the-art deep learning algorithms for Computer Vision with the ability to relate and contrast different concepts.</p>	
Inhalt	<p>Methods in Computer Vision based on Deep Learning, esp.:</p> <ul style="list-style-type: none"> ? deep learning in Computer Vision and convolutional neural networks ? generative/discriminative methods ? supervised/unsupervised methods, reinforcement learning ? video analysis ? Object and action recognition ? local and global feature extraction ? hierarchical object representations ? novel applications 	
Voraussetzungen	recommended are: solid programming skills and knowledge of basic calculus, statistics, and linear algebra	
Pruefungsmodalitaeten	project work with final report or oral/written examination	
Vergabe der LP	Bestehen der Modulprüfung	

Nuetzliche Literatur	Will be presented in class
---------------------------------	----------------------------

Discrete Structures 1

Code IDS1	Name Discrete Structures 1	
Leistungspunkte 8 LP	Dauer one semester	Turnus every winter semester
Lehrform Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand 240 h; thereof 90 h lecture 20 h preparation for exam 130 h self-study and working on assignments/projects (optionally in groups)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, B.Sc. Mathematik
Lernziele	<p>Students</p> <ul style="list-style-type: none"> - understand several basic graph parameters and the central theorems in these areas - can solve easy problems involving discussed topics - can describe graph algorithms computing discussed graph parameters - know how to use graphs and graph parameters to model real world problems 	
Inhalt	<ul style="list-style-type: none"> - Introduction to graph theory terminology - Matchings in graph and hypergraphs - Graph connectivity - Planar graphs - Graph Colouring - Hamilton Cycles - Ramsey Theory - Random graphs - Algebraic Graph constructions (Cayley graphs, Kneser graphs,...) - Algorithms computing discussed graph parameters 	
Voraussetzungen	recommended are: Einführung in die Praktische Informatik (IPI), Mathematik für Informatiker 1 (IMI1) oder Lineare Algebra 1 (MA4), Mathematik für Informatiker 2 (IMI2) oder Analysis 1 (MA1)	
Prüfungsmodalitäten	Assignments; at least 50% of the credit points for the assignments need to be obtained to be eligible to participate in the final exam; final written or oral exam (to be determined by the lecturer)	
Vergabe der LP	Bestehen der Modulprüfung	

Nuetzliche Literatur	<ul style="list-style-type: none">- Reinhard Diestel Graph Theory, 5th edition, Springer, 2016/17- Douglas West, Introduction to Graph Theory, Pearson, 2011.- J.A. Bondy and U.S.R. Murty, Graph Theory, Springer, 2008.- Bernhard Korte and Jens Vygen, Combinatorial Optimization, 6th edition, 2018.
---------------------------------	---

Discrete Structures 2

Code IDS2	Name Discrete Structures 2	
Leistungspunkte 8 LP	Dauer one semester	Turnus every summer semester
Lehrform Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand 240 h; thereof 90 h lecture 20 h preparation for exam 130 h self-study and working on assignments/projects (optionally in groups)	Verwendbarkeit M.Sc. Angewandte Informatik, M.Sc. Mathematik
Lernziele	<p>Students</p> <ul style="list-style-type: none"> - understand several advanced graph parameters and the central theorems in these areas - can solve problems involving discussed topics - can reprove the central considered results 	
Inhalt	<ul style="list-style-type: none"> - Probabilistic Methods - Extremal graph theory - Expander graphs - Quasirandom graphs - Further advanced topics 	
Voraussetzungen	recommended is: Discrete Structures 1	
Pruefungs- modalitaeten	Assignments; at least 50% of the credit points for the assignments need to be obtained to be eligible to participate in the final exam; final written or oral exam (to be determined by the lecturer)	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur	<ul style="list-style-type: none"> - Reinhard Diestel Graph Theory, 5th edition, Springer, 2016/17 - Douglas West, Introduction to Graph Theory, Pearson, 2011. - J.A. Bondy and U.S.R. Murty, Graph Theory, Springer, 2008. - Bernhard Korte and Jens Vygen, Combinatorial Optimization, 6th edition, 2018. 	

Formale Sprachen und Automatentheorie

Code IFSA	Name Formale Sprachen und Automatentheorie	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus mindst. jedes 4. Semester
Lehrform Vorlesung 4 SWS, Übungen 2 SWS mit Hausaufgaben.	Arbeitsaufwand 240 h; davon 90 h Präsenzstudium 60 h Prüfungsvorbereitung (und Prüfung) 90 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik

Lernziele	<p>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 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, können die Klasse der kontextfreien Sprachen durch kontextfreie Grammatiken und durch Kellerautomaten beschreiben und die Äquivalenz dieser Darstellungen beweisen, 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, 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	<p>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.</p>
Voraussetzungen	<p>empfohlen ist: Einführung in die Theoretische Informatik (ITH)</p>
Prüfungsmodalitäten	<p>eine mündliche Abschlussprüfung.</p>
Vergabe der LP	<p>Bestehen der Modulprüfung</p>

Nuetzliche Literatur	
---------------------------------	--

Fortgeschrittenenpraktikum

Code IFP	Name Fortgeschrittenenpraktikum	
Leistungspunkte 8 LP	Dauer	Turnus jedes Semester
Lehrform Praktikum 6 SWS	Arbeitsaufwand 240 h; davon mind. 25 h Präsenzzeit 10 h Vorbereitung Vortrag	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik
Lernziele	<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 projektypischen Kompetenzen vertieft, insbesondere das Arbeiten im Team (von bis zu drei Studierenden):</p> <p>Durchführung und Evaluation von Projekten und ihrer Phasenstruktur</p> <p>Planung und Durchführung von Projekt- und Teamarbeit.</p> <p>Zu den zu trainierenden Softskills zählen somit insbesondere Teamfähigkeit, Verfeinerung von Präsentationstechniken, etwaige Erschließung wissenschaftlicher Literatur sowie eigenverantwortliches Arbeiten.</p>	
Inhalt	<p>Domänenkenntnisse abhängig von den DozentInnen; allgemeine Lerninhalte sind:</p> <p>Vertiefung in die Projektarbeit</p> <p>Eigenständige Entwicklung von komplexer Software und deren Dokumentation</p>	
Voraussetzungen	empfohlen sind: Anfängerpraktikum (IAP), Einführung in Software Engineering (ISW)	
Pruefungs-modalitaeten	Bewertung der dokumentierten Software, des Projektberichts (5-10 Seiten) und des Vortrags (ca. 30 Minuten zzgl. Diskussion)	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur		

Fundamentals of Machine Learning

Code IFML	Name Fundamentals of Machine Learning	
Leistungspunkte 8 LP	Dauer one semester	Turnus in (irregular) alternation with *Machine Learning*
Lehrform 4 SWS lecture (in English), 2 SWS tutorial, homework assignments	Arbeitsaufwand 240h; thereof 90h lectures and tutorials 90h lecture wrap-up and homework 60h graded final report	Verwendbarkeit cannot be combined with *Machine Learning* B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	Students understand fundamental concepts of machine learning (features vs. response, unsupervised vs. supervised training, regression vs. classification etc.), get to know established learning methods and algorithms, are able to apply them to real-world problems, and can objectively assess the quality of the results. In addition, students learn how to use Python-based machine learning software such as scikit-learn.	
Inhalt	The lecture, along with its sibling *Advanced Machine Learning*, offers an extended version of the one-semester course *Machine Learning*, with more room for regression methods, unsupervised learning and algorithmic details: Classification (nearest neighbor rules, linear and quadratic discriminant analysis, logistic regression, classical and randomized decision trees, support vector machines, ensemble methods); regression (linear and non-linear least squares, regularized and sparse regression, robust regression); unsupervised learning (hierarchical clustering, k-means algorithm, Gaussian mixture models and expectation maximization, principal component analysis, non-linear dimension reduction); evaluation (risk minimization, model selection, cross-validation)	
Voraussetzungen	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
Pruefungsmodalitaeten	cannot be combined with *Machine Learning*, written exam (report on a 60h mini-research project)	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur	Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning (2nd edition), Springer, 2009	

Geometric Modeling and Animation

Code IGMA	Name Geometric Modeling and Animation	
Leistungspunkte 8 LP	Dauer one semester	Turnus every 3rd semester
Lehrform Lecture 4 SWS, Exercise 2 SWS	Arbeitsaufwand 240 h; thereof 90 h on-campus program 15 h exam preparation 135 h independent study and exercises (possibly in groups)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	<p>The students</p> <p>know the mathematical foundations of geometric modeling</p> <p>know the mathematical and physical foundations of computer animation</p> <p>know the algorithms and implementation aspects</p> <p>are familiar with the basics of animated movies</p> <p>are able to apply existing tools for geometric modeling and animation</p>	
Inhalt	<p>Introduction to curves</p> <p>Interpolating curves</p> <p>Bézier curves</p> <p>B-Splines</p> <p>Rational curves</p> <p>Introduction to surfaces</p> <p>Tensor product surfaces</p> <p>Transfinite surfaces and extrusion</p> <p>Subdivision</p> <p>Subdivision surfaces</p> <p>Animation and simulation</p> <p>Rigid body kinematics</p> <p>Particle systems</p> <p>Mass-spring models</p> <p>Cloth modeling</p> <p>Numerical methods for differential equations</p> <p>Collision detection and handling</p> <p>Fluid simulation and natural phenomena</p>	
Voraussetzungen	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
Prüfungsmodalitäten	a written or oral exam	

Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	<ul style="list-style-type: none"> - Curves and Surfaces for CAGD ? A Practical Guide, G. Farin, Morgan Kaufmann, 2002 - Computer Animation ? Algorithms and Techniques, R. Parent, Morgan Kaufmann, 2002 - 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, D. Eberly, Morgan Kaufmann, 2000 - Graphische Datenverarbeitung I, J. Encarnacao, W. Straßer, R. Klein, 4. Auflage, Oldenbourg 1996 - Advanced Animation and Rendering Techniques, A. Watt, M. Watt, Addison-Wesley, 1992 - Grundlagen der geometrischen Datenverarbeitung, J. Hoschek, D. Lasser, Teubner 1992 - Numerical Recipes ? The Art of Scientific Computing, W.H. Press, P. Flannery, S.A. Teukolsky, W.T. Vetterling, Cambridge University Press, 1986

Hardware Aware Scientific Computing

Code IHASC	Name Hardware Aware Scientific Computing	
Leistungspunkte 8 LP	Dauer one semester	Turnus irregular
Lehrform Lecture 4 SWS + Exercise Course 2 SWS	Arbeitsaufwand 240h;thereof 90h lecture 15h preparation for exam 135h self-study and working on assignments/projects (optionally in groups)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	Students are familiar with different forms of parallelism in modern computer architectures. They can exploit this parallelism selecting an appropriate programming model. They are familiar with modelling of parallelism and know fundamental parallel algorithms from scientific computing.	
Inhalt	<p>Parallel Computer Architecture</p> <ul style="list-style-type: none"> - Pipelining and super-scalar processors, SIMD vectorisation - Caches - Multicore architectures - GPUs - Communication networks <p>Programming Models</p> <ul style="list-style-type: none"> - Shared memory programming with OpenMP and C++ threads - OpenCL or Cuda - Task-based programming - Message-passing, MPI <p>Parallel Algorithms</p> <ul style="list-style-type: none"> - Speedup & scalability - Roofline model - Linear Algebra: Matrix-Vector, Matrix multiplication, solving dense systems, solving sparse systems - Iterative Solution of Linear Systems - High-Performance Libraries - Differential equations - Particle Methods 	
Voraussetzungen	basic knowledge in computer architecture and numerical methods; good programming skills in C++	

Pruefungs- modalitaeten	4-6 assignments highlighting the material covered in the lecture; assignments can be worked on in groups of 2-3 students with clear responsibilites
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	Frédéric Magoules, François-Xavier Roux, Guillaume Houzeaux: Parallel Scientific Computing, Wiley, 2016, doi: 10.1002/9781118761687

Inverse Problems

Code IIP	Name Inverse Problems	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus jedes Sommersemester
Lehrform Vorlesung 2 SWS, Übung 2 SWS und Hausarbeiten	Arbeitsaufwand 240 h; davon 60 h Präsenzstudium 15 h Prüfungsvorbereitung 165 h Selbststudium und Aufgabenbearbeitung, Hausarbeiten	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
Lernziele	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	
Voraussetzungen	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Numerische Mathematik	
Pruefungs- modalitaeten	eine mündliche Abschlussprüfung	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur	M. Bertero, P. Boccacci: Introduction to Inverse Problems in Imaging, IoP, 2002 web-Page and book: http://www.slaney.org/pct/pct-toc.html	

IT-Projektmanagement

Code IPM	Name IT-Projektmanagement	
Leistungspunkte 3 LP als Spezialvorlesung Angewandte Informatik 6 LP als EPG	Dauer ein Semester	Turnus jedes 2. Wintersemester
Lehrform Vorlesung+Übung 2 SWS für eine Anrechnung als EPG mit zusätzlichem Forschungsprojekt	Arbeitsaufwand Vorlesung + Übung 90 h insgesamt, davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen) für eine Anrechnung als EPG zusätzlich: 90 h Forschungsprojekt, davon 15 h Einarbeitung und Literaturrecherche 15 h Vorbereitung einer Studie, z.B. Interview-Training 20 h Durchführung einer Studie 20 h Auswertung 20 h Abschlusspräsentation + Berichterstellung	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
Lernziele	Die Teilnehmer/innen können ein Projekt planen und überwachen, verstehen, wie Projekte in Organisationen eingebettet sind und haben Grundkenntnisse in vertraglichen Themen. Die Teilnehmer/innen des EPG können ein Forschungsprojekt selbständig durchführen und kennen Forschungsergebnisse aus der Gender-Forschung.	

Inhalt	Projektplanung, Projektorganisation Kostenschätzung Angebot/ Vertrag, Verhandeln Vorgehensmodelle Risikomanagement Controlling IT-Vertragsrecht Änderungsmanagement Zeitmanagement Projektabschluss Verteilte Softwareentwicklung
Voraussetzungen	keine
Pruefungs- modalitaeten	eine Abschlussprüfung Für EPG: Zusätzlich erfolgreiche Teilnahme am Forschungsprojekt. Die Note für das Forschungsprojekt geht zur Hälfte in die Endnote ein.
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	PMI (Project Management Institute): A Guide to the Project Management Body of Knowledge (PM BOK ® Guide), 4. Ausgabe 2008

IT-Sicherheit

Code IITS	Name IT-Sicherheit	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus unregelmäßig
Lehrform Vorlesung 4 SWS + Übung 2 SWS	Arbeitsaufwand 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
Lernziele	<p>Studierende</p> <ul style="list-style-type: none"> * erwerben umfangreiches Wissen über die Funktionsweise und Verwundbarkeiten vernetzter Computersysteme, und können somit Konzepte zur IT-Netzsicherheit bewerten und entwerfen * erlangen grundlegende Kenntnisse über die Sicherung großer Netzwerke und der Kommunikationsinfrastruktur (Routing, Namensauflösung, Internet-Firewalls, Intrusion Detection Systeme) * erwerben Kenntnisse im Bereich Kryptographie: Theorie der Kryptographie und praktische Umsetzung typischer kryptographischer Verfahren im Zusammenhang mit kryptographischen Prüfverfahren, symmetrischen und asymmetrischen Chiffrierverfahren * erwerben umfassende Kompetenzen zur Detektion von Cyberangriffen; grundlegende Kompetenzen im Feld der IT-/Cyber-Forensik * erwerben praktische Erfahrungen bei der Verwendung von dedizierter Software zur Detektion von Angriffsszenarien im Datennetz * erwerben grundlegender Kenntnisse zum BSI IT-Grundschutzkonzept * erwerben grundlegender Kenntnisse im Penetrationstesting und dazugehöriger Werkzeuge <p>Langfristiges Ausbildungsziel: Einsatz-/Beschäftigungsfähigkeit in der Breite des Arbeitsfeldes IT-Sicherheit;</p>	

Inhalt	<p>Der IT-Sicherheit kommt bei der allgegenwärtigen Digitalisierung eine Schlüsselrolle zu. Diese Vorlesung vermittelt methodische Ansätze zur Modellierung und Bewertung von Angriffsszenarien, auf Basis welcher wirksame technische Gegenmaßnahmen umgesetzt werden können. Insbesondere werden folgende Schwerpunkte adressiert:</p> <ul style="list-style-type: none"> - Sicherheitsmodelle und Bewertungskriterien - Kryptographische Prüfwerte: Modifikationserkennungs- und Nachrichtenauthentisierungswerte - Symmetrische und asymmetrische kryptographische Verfahren - Kryptographische Protokolle - Zugangs/Zutritts/Zugriffskontrolle - Schutz von Kommunikationsinfrastrukturen - Digitale Identität - Penetrationstesting / Schwachstellenanalyse - BSI IT-Grundschutzbausteine <p>Mit Hilfe von virtuellen Maschinen in einem geschützten Bereich werden klassische Angriffs- und Schutzszenarien praktisch untersucht. Hier werden sog. Experimentierblätter samt Daten zur Verfügung gestellt.</p>
Voraussetzungen	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Betriebssysteme und Netzwerke (IBN), Einführung in die Technische Informatik (ITE), Algorithmen und Datenstrukturen (IAD)
Pruefungs-modalitaeten	Regelmäßige Teilnahme an der Vorlesung, Erfolgreiche Teilnahme an den Übungen (Erreichen von 50% der Punkte) und Bestehen der Abschlussklausur (90 Min.)
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	<p>C. Eckert, IT-Sicherheit: Konzepte, Verfahren, Protokolle De Gruyter Studium. Oldenbourg: de Gruyter, 2014.</p> <p>T.W. Harich, IT-Sicherheitsmanagement: Arbeitsplatz IT Security Manager. MITP, 2012.</p> <p>J.P.Müller, Security for Web Developers, O'Reilly, 2018</p>

Knowledge Discovery in Databases

Code IKDD	Name Knowledge Discovery in Databases	
Leistungspunkte 8 LP	Dauer one semester	Turnus every 2nd winter semester
Lehrform Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand 240 h; thereof 90 h lecture 20 h preparation for exam 130 h self-study and working on assignments/projects (optionally in groups)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	<p>Students</p> <ul style="list-style-type: none"> - understand the KDD process and when to apply different KDD tasks - are able to apply suitable data mining techniques to specific data analysis problem - know the foundations of statistics and probability theory underlying diverse data mining techniques - can apply and adopt different data clustering algorithms and models - can apply and adopt different data classification algorithms and models - understand different methods and metrics to evaluate the quality of data mining results - can describe different pattern detection methods to obtain frequent patterns from diverse types of data sets - are familiar with the foundations of models and techniques to extract patterns from graph data - can apply and realize the different algorithms and data mining procedures in software environments such as R or Python 	
Inhalt	<ul style="list-style-type: none"> - KDD process and tasks - Data, statistics, and probability theory - Clustering models, techniques, and algorithms - Classification models, techniques, and algorithms - Frequent pattern mining approaches - Outlier detection concepts - Graph mining models and methods 	
Voraussetzungen	recommended are: Algorithmen und Datenstrukturen (IAD), Effiziente Algorithmen 1 (IEA1), Einführung in die Wahrscheinlichkeitstheorie und Statistik (MA8)	

Pruefungs- modalitaeten	final written exam
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	<ul style="list-style-type: none"> - Jiawei Han, Micheline Kamber, and Jian Pei: Data Mining. Concepts and Techniques, Morgan Kaufmann Series in Data Management Systems, 2011 (3rd Edition). - Charu Aggarwal: The textbook. Springer, 2015. - Pang-Ning Tan, Michael Steinbach, and Vipin Kumar: Introduction to Data Mining. Addison Wesley, 2005.

Machine Learning

Code IML	Name Machine Learning	
Leistungspunkte 8 LP	Dauer one semester	Turnus in (irregular) alternation with *Fundamentals of Machine Learning* + *Advanced Machine Learning*
Lehrform 4 SWS lecture (in English), 2 SWS tutorial, homework assignments	Arbeitsaufwand 240h; thereof 90h lectures and tutorials 120h lecture wrap-up and homework 30h preparation for examination	Verwendbarkeit cannot be combined with *Fundamentals of Machine Learning* or *Advanced Machine Learning* B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	Students understand a broad range of machine learning concepts, get to know established and advanced learning methods and algorithms, are able to apply them to real-world problems, and can objectively assess the quality of the results. In addition, students learn how to use Python-based machine learning software such as scikit-learn.	
Inhalt	This lecture is a compact version of the two-semester course *Fundamentals of Machine Learning* + *Advanced Machine Learning*: Classification (linear and quadratic discriminant analysis, neural networks, linear and kernelized support vector machines, decision trees and random forests), least squares and regularized regression, Gaussian processes, unsupervised learning (density estimation, cluster analysis, Gaussian mixture models and expectation maximization, principal component analysis, bilinear decompositions), directed probabilistic graphical models, optimization for machine learning, structured learning	
Voraussetzungen	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
Pruefungsmodalitaeten	cannot be combined with *Fundamentals of Machine Learning* or *Advanced Machine Learning*, successful homework solutions (at least 50% of total achievable points) and oral examination	
Vergabe der LP	Bestehen der Modulprüfung	

Nuetzliche Literatur	Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning (2nd edition), Springer, 2009; David Barber: Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012
---------------------------------	---

Mining Massive Datasets

Code IMMD	Name Mining Massive Datasets	
Leistungspunkte 6 LP	Dauer one semester	Turnus at least every 4th semester
Lehrform Lecture 2 SWS + Exercise course 2 SWS	Arbeitsaufwand 180 h; thereof 60 h lecture 15 h preparation for exam 105 h self-study and working on assignments (optionally in groups)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	<ul style="list-style-type: none"> * Knowledge of selected approaches and programming paradigms of parallel data processing * Knowledge how to use tools for parallel data processing (among others Apache Hadoop and Spark) * Familiarity with application domains of big data analysis * Knowledge of methods of parallel pre-processing of data * Knowledge of methods like classification, regression, clustering and their parallel implementations * Knowledge of scaling of parallel algorithms 	
Inhalt	<p>This module covers the following topics:</p> <ul style="list-style-type: none"> * programming paradigms for parallel-distributed data processing, especially Map-Reduce and Spark programming models * usage of tools like Apache Spark, Hadoop, Pig, Hive, and possibly other frameworks for parallel-distributed data processing * application cases in parallel data analysis, for example clustering, recommendation, search for similar objects, mining of data streams * techniques for parallel pre-processing of data * fundamentals of analysis techniques such as classification, regression, clustering and evaluation of the results * parallel algorithms for data analysis and their implementations * theory and practice of scalability and tuning of frameworks 	
Voraussetzungen	recommended are Knowledge of Java/Python and in elementary probability theory / statistics; module IBD can be taken as a complement / extension.	
Pruefungsmodalitaeten	a written exam	

Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	<ul style="list-style-type: none"> * Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, Mining of Massive Datasets, Cambridge University Press, Version 2.1 von 2014 (http://www.mmds.org/) * Trevor Hastie, Robert Tibshirani, Jerome Fried-man, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009 (http://statweb.stanford.edu/~tibs/ElemStatLearn/) * Ron Bekkerman, Misha Bilenko, John Langford, Scaling Up Machine Learning, Cambridge University Press, 2012 * Jiawei Han, Micheline Kamber, Jian Pei, Data Mining: Concepts and Techniques, Morgan Kaufmann, (third edition), 2012 * Books from O'Reilly Data Science Starter Kit, 2014 (http://shop.oreilly.com/category/get/data-science-kit.do)

Object-Oriented Programming for Scientific Computing

Code IOPSC	Name Object-Oriented Programming for Scientific Computing	
Leistungspunkte 6 LP	Dauer one semester	Turnus every summer semester
Lehrform Lecture 2 SWS, Exercise on computer 2 SWS	Arbeitsaufwand 180 h; thereof 60 h lecture 105 h self-study and working on assignments 15 h preparation for exam	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	The students are proficient in the programming language C++, can assess the performance of different programming techniques, know template programming techniques, and can use the Standard Template Library (STL). They can apply their new skills to solve selected problems of Scientific Computing.	
Inhalt	This module deepens the skills in object-oriented programming obtained in the basic lecture Einführung in die Praktische Informatik (IPI) with special emphasis on Scientific Computing: Class concept Dynamic memory allocation Exception handling Resource allocation and initialization Constness Static versus dynamic polymorphism Traits and Policies Standard Template Library Template Metaprogramming Parallel programming techniques	
Voraussetzungen	recommended are: Einführung in die Praktische Informatik (IPI), alternatively basic knowledge of an object-oriented programming language	
Pruefungs- modalitaeten	a written exam at the end of the semester	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur		

Object Recognition and Image Understanding

Code IORIU	Name Object Recognition and Image Understanding	
Leistungspunkte 8 LP	Dauer one semester	Turnus at least every 4th semester
Lehrform Lecture 4 h + Exercise course 2 h	Arbeitsaufwand 240h; thereof 90h lectures and tutorials 100h lecture wrap-up and homework 50h preparation for project work, examination	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik M.Sc. Scientific Computing
Lernziele	<p>To have reached understanding of the state-of-the-art in mid- and high-level Computer Vision and to have the ability to relate and contrast different concepts.</p> <p>To be able to apply essential algorithms from pattern recognition and deep learning to current problems in machine vision.</p> <p>To be capable of understanding and analyzing the latest publications in Computer Vision and to evaluate their strengths and weaknesses.</p> <p>To have a firm command of the algorithmic basics and to be able to analyze and solve object recognition problems in novel application areas.</p> <p>To know the most relevant methods for robust object representation and to judge them based on their applicability and restrictions.</p>	
Inhalt	<p>Methods in mid- and high-level Computer Vision, esp.:</p> <ul style="list-style-type: none"> - object detection and classification - deep learning in Computer Vision and convolutional neural networks - machine learning approaches for object representation - video analysis - recognition of human actions - local and global feature extraction - model based approaches - view based approaches - generative/discriminative methods - supervised/unsupervised methods - registration - shape analysis - voting and hashing methods - hierarchical object representations 	

Voraussetzungen	recommended are: solid programming skills and knowledge of basic calculus, statistics, and linear algebra
Pruefungs-modalitaeten	project work with final report or oral/written exam
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	Will be presented in class

Optimization for Machine Learning

Code IOML	Name Optimization for Machine Learning	
Leistungspunkte 8 LP	Dauer one semester	Turnus every winter semester
Lehrform Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand 240 h; thereof 60 h lectures 30 h exercises 24 h preparation for exam 126 h self-study and working on assignments/projects (optionally in groups)	Verwendbarkeit B.Sc. Angewandte Informatik M.Sc. Angewandte Informatik M.Sc. Mathematik M.Sc. Scientific Computing
Lernziele	<p>The students</p> <ul style="list-style-type: none"> - can analyze optimization methods for machine learning problems and estimate the area of their potential application - can competently apply existing algorithms and program packages for inference and learning with graphical models and neural networks - know typical optimization techniques for inference and learning with graphical models and neural networks - understand the basics of convex analysis, convex optimization, convex duality theory, (integer) linear programs and their geometry 	

Inhalt	<p>The course presents various existing optimization techniques for such important machine learning tasks, as inference and learning for graphical models and neural networks. In particular, it addresses such topics as combinatorial algorithms, integer linear programs, scalable convex and non-convex optimization and convex duality theory. Graphical models and neural networks play a role of working examples along the course. The content of the course includes:</p> <ul style="list-style-type: none"> - Convex analysis and optimization: convex sets and functions, polyhedra, (integer) linear programs, basic first-order convex optimization methods and their stochastic variants, LP and Lagrange relaxations - Graphical Models: dynamic programming, sub-gradient and block-coordinate ascent inference methods, min-cut/max-flow based inference, structured risk minimization for graphical models - neural networks: architectures, backpropagation algorithm, stochastic gradient descent and its variants for training neural networks.
Voraussetzungen	recommended are: linear algebra, analysis and any universal programming language (e.g. C/C++/Pascal/python)
Pruefungsmodalitaeten	Oral or written exam (is defined depending the number of students at the beginning of the course).
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	Will be given by lecturer at the beginning of the course

Praktische Geometrie

Code IPGeo	Name Praktische Geometrie	
Leistungspunkte 4 LP	Dauer ein Semester	Turnus unregelmäßig
Lehrform Vorlesung 2 SWS, Übung 1 SWS	Arbeitsaufwand 120 h; davon 45 h Präsenzstudium 60 h Aufgabenbearbeitung 15 h Prüfungsvorbereitung	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	Verständnis grundlegender geometrischer Konzepte zur Datenanalyse sowie effektive Punktsuche und Weiterverarbeitung von Messdaten Souveräner Umgang mit Projektionen und Beschreibungen jenseits der dreidimensionalen Erfahrungswelt Berechnung geometrischer Invarianten, Distanzen, Krümmungen aus Messdaten, rekonstruierten und generierten Flächen	
Inhalt	Grundlegende Gebiete der Geometrie mit Relevanz in Computergraphik, Bildverarbeitung, Mustererkennung, Computer Vision und Geometrischem Modellieren (i) Analytische Geometrie: Operationen auf Vektorräumen mit geeigneten Koordinaten und Abbildungen (Affinitäten, Kollinearitäten), geometrische Ausgleichsprobleme aus fehlerbehafteten Messdaten (ii) Projektive Geometrie: Zentralprojektion und inverse Rekonstruktion von 3D-Objekten aus ebenen Bildern (Computer Vision, Geodäsie), Unterschiede zwischen B-Spline-Kurven und -Flächen und der Klasse der NURBS, Freiformgeometrien in CAD-Systemen (iii) Differentialgeometrie: Parameterdarstellungen in der geometrischen Datenverarbeitung, implizite Darstellungen (level sets), Abschätzung von Invarianten aus diskreten Daten (Triangulierungen, Punktwolken)	
Voraussetzungen	empfohlen sind: Einführung in die Praktische Informatik (IPI), Mathematik für Informatiker (IMI1 und 2) oder Lineare Algebra (MA4)	
Pruefungs-modalitaeten	eine schriftliche oder mündliche Abschlussprüfung	
Vergabe der LP	Bestehen der Modulprüfung	

Nuetzliche Literatur	Geometrie für Informatiker, Skript TU Wien 2004, Helmut Pottmann Aktuelle Fachveröffentlichungen
---------------------------------	---

Projektseminar Biomedizinische Bildanalyse

Code IPBB	Name Projektseminar Biomedizinische Bildanalyse	
Leistungspunkte 6 LP	Dauer ein Semester	Turnus jedes Sommersemester
Lehrform 2 Teile Seminar und Projekt, 4 SWS	Arbeitsaufwand 180 h (je zur Hälfte Seminar und Projekt) 60 h Präsenzstudium 120 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
Lernziele	<p>Die Studierenden erlangen vertiefte Kenntnisse und Fähigkeiten im Gebiet Biomedizinische Bildanalyse</p> <p>lernen fortgeschrittene Methoden und Algorithmen zur automatischen Analyse biomedizinischer Bilder</p> <p>lernen wie man Algorithmen und Software für automatische Bildanalyse entwickelt</p> <p>erweitern ihre Fähigkeiten Projektergebnisse mündlich zu präsentieren und schriftlich zu dokumentieren</p> <p>erweitern ihre Fähigkeiten zur Teamarbeit und zur Strukturierung von Projekten</p>	
Inhalt	<p>Die Studierenden arbeiten in Teams an ausgewählten fortgeschrittenen Themen der Biomedizinischen Bildanalyse. Der Schwerpunkt liegt auf der automatischen Analyse von Zellmikroskopiebildern und medizinischen tomographischen Bildern. Beispiele für Themen sind die Segmentierung und Verfolgung (Tracking) von Zellen in Mikroskopiebildern, die Segmentierung von Blutgefäßen in tomographischen Bildern sowie die Registrierung von Magnetresonanz (MR) Bildern des menschlichen Gehirns. Die Veranstaltung besteht aus einem Seminarteil (Einarbeitung in die relevante Literatur, Erarbeitung der theoretischen Grundlagen, Vortragspräsentation) und einem Projektteil (Spezifikation eines Softwaresystems, Entwurf von Algorithmen und Implementierung von Bildanalyseverfahren, Test und Evaluierung der Verfahren, Präsentation der Ergebnisse).</p>	
Voraussetzungen	empfohlen sind: Grundkenntnisse in Bildverarbeitung (Computer Vision, Image Analysis), Programmierkenntnisse, Kenntnisse in Software Engineering	

Pruefungs- modalitaeten	Vortragspraesentationen von Zwischen- und Endergebnissen (jeder Studierende 4 Vortraege je ca. 10 Min. und anschließender Diskussion) Schriftliche Ausarbeitung der theoretischen Grundlagen, der verwendeten Methoden und der Ergebnisse (jeder Studierende ca. 10 Seiten)
Vergabe der LP	Bestehen der Modulpruefung
Nuetzliche Literatur	Bekanntgabe in der Lehrveranstaltung

Qualitätsmanagement

Code ISWQM	Name Qualitätsmanagement	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus jedes 2. Sommersemester
Lehrform Vorlesung 3 SWS, Übung 3SWS	Arbeitsaufwand 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung im Team	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
Lernziele	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	
Voraussetzungen	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden	
Prüfungsmodalitäten	eine mündliche oder schriftliche Abschlussprüfung	
Vergabe der LP	Bestehen der Modulprüfung	
Nützliche Literatur	Wird jährlich aktualisiert	

Randomisierte Algorithmen

Code IRA	Name Randomisierte Algorithmen	
Leistungspunkte 6 LP	Dauer ein Semester	Turnus mindst. jedes 4. Semester
Lehrform Vorlesung 3 SWS, Übung 1 SWS	Arbeitsaufwand 180 h; davon 60 h Präsenzstudium 40 h Prüfungsvorbereitung 80 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	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	
Voraussetzungen	empfohlen sind: elementare Grundkenntnisse in Algorithmen wie sie z.B. im Modul Algorithmen und Datenstrukturen (IAD) vermittelt werden.	
Prüfungsmodalitäten	eine mündliche oder schriftliche Abschlussprüfung	
Vergabe der LP	Bestehen der Modulprüfung	

Nuetzliche 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.
---------------------------------	---

Requirements Engineering

Code ISWRE	Name Requirements Engineering	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus jedes 2. Sommersemester
Lehrform Vorlesung 3 SWS, Übung 3SWS	Arbeitsaufwand 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung im Team	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
Lernziele	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)	
Voraussetzungen	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden	
Pruefungs- modalitaeten	eine mündliche oder schriftliche Abschlussprüfung	
Vergabe der LP	Bestehen der Modulprüfung	
Nuetzliche Literatur	Wird jährlich aktualisiert	

Scientific Visualization

Code ISV	Name Scientific Visualization	
Leistungspunkte 8 LP	Dauer one semester	Turnus every 3rd semester
Lehrform Lecture 4 SWS, Exercise 2 SWS	Arbeitsaufwand 240 h; thereof 90 h on-campus program 15 h exam preparation 135 h independent study and exercises (possibly in groups)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	The students understand fundamental and advanced concepts of scientific visualization. They understand the mathematical fundamentals, data structures, and implementation aspects. They get to know schemes for interpolation and integration, mapping for scalar, vector, and tensor fields, and derived approaches. The students understand approaches for direct and indirect volume rendering, feature extraction, and topology-based analysis. The students are able to apply these concepts to real-world problems using existing software packages, and develop small programs using visualization libraries.	
Inhalt	<ul style="list-style-type: none"> - Introduction - Visualization Process - Data Sources and Representation - Interpolation and Filtering - Approaches for Visual Mapping - Scalar Field Visualization: Advanced Techniques for Contour Extraction, Classification, Texture-Based Volume Rendering, Volumetric Illumination, Advanced Techniques for Volume Visualization, Pre-Integration, Cell Projection, Feature Extraction - Vector Field Visualization: Vector Calculus, Particle Tracing on Grids, Vector Field Topology, Vortex Visualization, Feature Extraction, Feature Tracking - Tensor Field Visualization: Glyphs, Hue-Balls and Lit-Tensors, Line-Based Visualization, Tensor Field Topology, Feature Extraction 	
Voraussetzungen	strongly recommended is: Computer Graphics (ICG) recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
Prüfungsmodalitäten	an oral or written exam	

Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	C.D. Hansen, C.R. Johnson, The Visualization Handbook, 2005.

Software Evolution

Code ISWEvol	Name Software Evolution	
Leistungspunkte 3 LP	Dauer ein Semester	Turnus unregelmäßig
Lehrform Vorlesung 2 SWS	Arbeitsaufwand 90h; davon 30 h Vorlesung 35 Aufgabenbearbeitung und Aufarbeitung/Selbststudium 25 h Prüfungsvorbereitung (evtl. in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
Lernziele	<p>Die Studierenden können nach der Vorlesung: ein Software-Reengineering-Projekt fachlich planen und beurteilen, bei der Ersterstellung von Software die Evolutionsfähigkeit konzeptuell sicherstellen, ein Wartungskonzept für eine erstellte Software aufbauen.</p> <p>Sie kennen 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, die klassischen Techniken der Softwaresanierung, die Typologie der Softwarewartung und das Management der Fehlerbehebung, die Relevanz der Thematik in der Praxis der industriellen Softwareerstellung.</p>	

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>
Voraussetzungen	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden
Pruefungs-modalitaeten	Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Anzahl der Teilnehmer)
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	<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 Anwendungs-systeme. dpunkt, Heidelberg 2005</p> <p>Smith, D.D.: Designing Maintainable Software. Springer, Heidelberg 1999</p>

Software Ökonomie

Code ISWÖk	Name Software Ökonomie	
Leistungspunkte 3 LP	Dauer ein Semester	Turnus unregelmäßig
Lehrform Vorlesung 2 SWS	Arbeitsaufwand 90h; davon 30 h Vorlesung 35 Aufgabenbearbeitung und Aufarbeitung/Selbststudium 25 h Prüfungsvorbereitung (evtl. in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
Lernziele	<p>Die Studierenden können nach der Vorlesung grob den Preis und die Lizenzierung einer erstellten Software ermitteln, die Vermarktung von Software planen und anstoßen, grob die Bilanz sowie die Gewinn- und Verlustrechnung eines Softwareherstellers verstehen, den Wert einer Software mit seinen verschiedenen Komponenten beurteilen, aus Sicht des Herstellers sowie aus Sicht der Nutzer, Preisverhandlungen zu Softwareprojekten planen. Sie kennen die Grundzüge der Kosten- und Leistungsrechnung (soweit sie für die Softwareerstellung relevant ist), die unterschiedlichen Vertragsarten, die im Umfeld der Softwareerstellung zum Einsatz kommen, die wichtigsten Verhandlungsstrategien bei der Verhandlung von Softwareverträgen, rechtliche Aspekte im Bereich der IT-Kriminalität, die Relevanz der Vorlesungsthemen in der Praxis der industriellen Softwareerstellung.</p>	

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 Begriffsklärung, Grundlagen Softwareökonomie Management von Softwareprojekten Wertermittlung von Software Bepreisung von Software Software-Marketing Verhandlungen und Verträge Bilanzierung und Rechnungslegung IT-Kriminalität Schadensabwendung Zusammenfassung</p>
Voraussetzungen	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden
Pruefungs-modalitaeten	eine mündliche oder schriftliche Abschlussprüfung (je nach Anzahl der TeilnehmerInnen)
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	<p>Buxmann, P.; Diefenbach, H.; Hess, T.: Die Softwareindustrie. Ökonomische Prinzipien, Strategien, Perspektiven. Heidelberg, 2008</p> <p>Herzwurm, G.; Pietsch, W.: Management von IT-Produkten. Heidelberg, 2009</p> <p>Mertens, P. (Hrsg.): Lexikon der Wirtschaftsinformatik, 3. Auflage, Heidelberg 2001</p> <p>Versteegen, G.: Marketing in der IT-Branche. Heidelberg 2003</p> <p>Wöhe, G.: Einführung in die Allgemeine Betriebswirtschaftslehre. München 2010</p>

Text Analytics

Code ITA	Name Text Analytics	
Leistungspunkte 8 LP	Dauer one semester	Turnus every 2nd winter semester
Lehrform Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand 240 h; thereof 90 h lecture 15 h preparation for exam 135 h self-study and working on assignments/projects (optionally in groups)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
Lernziele	<p>Students</p> <ul style="list-style-type: none"> - can implement and apply different text analytics methods using open source NLP and machine learning frameworks - can describe different document and text representation models and can compute and analyze characteristic parameters of these models - know how to determine, apply, and interpret use-case specific document similarity measures and underlying ranking concepts - know the concepts and techniques underlying different text classification and clustering approaches - know different models for phrase extraction and text summarization and are able to apply respective models and concepts using NLP and machine learning frameworks - know the fundamental methods for the extraction of document outlines at different levels of granularity - are familiar with basic concepts of topic models and their application in different text analytics tasks - understand the principles of evaluating results of text analytics tasks - know the theoretical background of machine learning methods at sufficient depths to be able to choose parameters and adapt an algorithm to a given text analytics problem - are aware of ethical issues arising from applying text analytics in different domains 	

Inhalt	<ul style="list-style-type: none"> - Text analytics in the context of Data Science - Open source text analytics, NLP, and machine learning frameworks - Fundamentals of NLP pipeline components - Document and text representation models - Document and text similarity metrics - Approaches, techniques and corpora for benchmarking text analytics tasks - Traditional and recent text classification and clustering approaches - Information extraction and topic detection approaches - Fundamentals of keyword and phrase extraction - Text summarization techniques - Generating document and text outlines - Ethical and legal aspects of text analytics methods - Text Analytics project management
Voraussetzungen	Recommended are: solid knowledge of basic calculus, statistics, and linear algebra; good Python programming skills
Pruefungs-modalitaeten	<p>Assignment (40%) and Programming Project (60%); about 4-5 assignments focusing on the material learned in class on a conceptual and formal level;</p> <p>group project in which 3-4 students develop a prototypical text analytics framework, including design and evaluation, a written project documentation as well as the code need to be submitted at the end of classes, clearly indicating what student is responsible for what part of the project.</p>
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	<p>The following textbooks and texts are useful but not required.</p> <ul style="list-style-type: none"> - Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft) - Yoav Goldberg. A Primer on Neural Network Models for Natural Language Processing (2015) - Christopher D. Manning and Hinrich Schütze: Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: May 1999. <p>Furthermore, during the course of this lecture, several papers covering topics discussed in class will be provided.</p>

Verteilte Systeme I

Code IVS1	Name Verteilte Systeme I	
Leistungspunkte 6 LP	Dauer ein Semester	Turnus mind. jedes 4. Semester
Lehrform Vorlesung 2 SWS, Übung 2 SWS	Arbeitsaufwand 180 h; davon 60 h Präsenzstudium 15 h Prüfungsvorbereitung 105 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
Lernziele	<p>Verständnis der unterschiedlichen parallelen Architekturen und der Besonderheiten von verteilten Systemen</p> <p>Kenntnis der grundlegenden theoretischen Probleme und Algorithmen in verteilten Systemen (z.B. Skalierbarkeit)</p> <p>Fähigkeit zur Erstellung von parallelen und verteilten Programmen, insbes. im Spark-Programmiermodell</p> <p>Kenntnis der praktischen Anwendung diverser Programmierparadigmen und Frameworks (Pthreads, MPI, Hadoop, Spark) für parallele oder verteilte Programmierung</p> <p>Vertrautheit mit skalierbarer Verarbeitung von Daten</p> <p>Kenntnisse über Probleme und Lösungen in Bereichen Fehlertoleranz und Verlässlichkeit der verteilten Systeme</p>	
Inhalt	<p>Das Modul behandelt die grundlegenden Prinzipien der parallelen und verteilten Systeme im Kontext ihrer Programmierung, insbesondere zum Zwecke der skalierbaren Verarbeitung von Daten. Es werden Konzepte aus den Bereichen Architekturen, Protokolle, Algorithmen, Implementierung und Softwareframeworks vorgestellt. Ein wesentlicher Teil der Vorlesung widmet sich der praktischen parallelen und verteilten Programmierung. Dazu gehörten u.a. Ansätze wie MPI, Map-Reduce, Spark-Programmiermodell und Actors. Ergänzende Themen umfassen Fehlertoleranz, effiziente Protokolle und Skalierbarkeit. Die Umsetzung in die Praxis erfolgt an Beispielen der Verarbeitung großer Datenmengen. Das Modul soll die Studierenden befähigen, Spezifika und Probleme der verteilten Systeme zu verstehen, und effiziente verteilte Anwendungen mit Softwareframeworks wie Apache Spark zu erstellen.</p>	
Voraussetzungen	empfohlen sind: Kenntnisse in Java (z.B. durch Einführung in Software Engineering (ISW)) oder Python	

Pruefungs- modalitaeten	eine Abschlussprüfung
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	<p>George Coulouris, Jean Dollimore, Tim Kindberg: Distributed Systems: Concepts and Design (4th ed.), Addison-Wesley, 2005.</p> <p>Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia: Learning Spark: Lightning-Fast Big Data Analysis, O'Reilly Media, 2015.</p> <p>Jure Leskovec, Anand Rajaraman, Jeff Ullman: Mining of Massive Datasets, 2nd edition (v2.1), Online: http://www.mmds.org/</p> <p>Grama, A., Gupta, A., Karypis, G., Kumar V.: Introduction to Parallel Computing, Addison-Wesley, 2. Auflage, 2003.</p> <p>Andrew S. Tanenbaum, Maarten Van Steen: Distributed Systems: Principles and Paradigms, Prentice Hall, 2006.</p>

Visualisierung im Bereich Cultural Heritage

Code IVCH	Name Visualisierung im Bereich Cultural Heritage	
Leistungspunkte 2 LP	Dauer ein Semester	Turnus unregelmäßig
Lehrform Vorlesung 2 SWS	Arbeitsaufwand 60 h; davon 30 h Präsenzstudium, 30 h Prüfungsvorbereitung	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
Lernziele	Die Studierenden sind mit unterschiedlichen Scantechniken vertraut und können Georadardaten interpretieren. Sie beherrschen den Umgang mit 3D Scan-, Georadar- und Magnetfelddaten, geophysikalischer Prospektion und weiteren Untersuchungen von Messdaten und kennen die Herangehensweise mit 2D und 3D Bildverarbeitung zur Erkennung von Merkmalen (Schrift). Sie wissen um die ethischen Grundsätze bei der Rekonstruktion, Befund und Hypothese (London Charter).	
Inhalt	Weißlicht- und Time-of-flight-Scanner, Rekonstruktionen von Gefäßen und Gebäuden, 3D-Puzzle, Skelettierung, ethische Grundsätze	
Voraussetzungen	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Computergraphik 1 (ICG1)	
Prüfungsmodalitäten	eine mündliche oder schriftliche Abschlussprüfung (je nach Teilnehmerzahl)	
Vergabe der LP	Bestehen der Modulprüfung	
Nützliche 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 Dokumentation und Klassifikation archäologischer Gefäße. Dissertation, TU Chemnitz, 2011	

Volumenvisualisierung

Code IVV	Name Volumenvisualisierung	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus jedes Sommersemester
Lehrform Vorlesung 2 SWS, Übung 3 SWS	Arbeitsaufwand 240 h; davon 75 h Präsenzstudium 15 h Prüfungsvorbereitung 150 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
Lernziele	<p>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 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>	
Voraussetzungen	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	

Pruefungs- modalitaeten	eine schriftliche oder mündliche Abschlussprüfung
Vergabe der LP	Bestehen der Modulprüfung
Nuetzliche Literatur	Engel et al: Real-Time Volume Graphics www.real-time-volume-graphics.org , Schroeder et al: VTK Textbook http://www.kitware.com/products/books/vtkbook.html

Wissensmanagement und Entscheidungen im Software Engineering

Code ISWKM	Name Wissensmanagement und Entscheidungen im Software Engineering	
Leistungspunkte 3 LP	Dauer ein Semester	Turnus jedes 2. Wintersemester
Lehrform Vorlesung+Übung 2 SWS	Arbeitsaufwand 90 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
Lernziele	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	Wissensmanagement Ontologien Rationale Re-engineering learning organization Entscheidungen Management-Entscheidungen, Business Case Risikomanagement Anforderungspriorisierung Entscheidungen im Entwurf: ATAM, SAAM, CBAM Entscheidungstheorie Entscheiden unter Ungewissheit Mathematical Economics Entscheidung mit mehreren Parteien: Harvard-Konzept, Verhandlungen Spieltheorie Fehlentscheidungen/ Decision Traps/ Biases	
Voraussetzungen	empfohlen sind: Vorlesung und Übung Einführung in Software Engineering (ISW) oder vergleichbare Vorkenntnisse	
Prüfungsmodalitäten	eine Abschlussprüfung.	
Vergabe der LP	Bestehen der Modulprüfung	

Nuetzliche Literatur	Raiffa, Howard; Richardson, John; Metcalfe, David: Negotiation analysis - the science and art of collaborative decision making, Belknap, Cambridge, 2002
---------------------------------	--

2.4 Module aus dem B.Sc./M.Sc. Mathematik

Aus dem Bachelor Mathematik mit 100% Fachanteil sind folgende Module anrechenbar:

- Wahrscheinlichkeitstheorie (MC4)
- Numerik (MD1)
- Statistik (MD2)
- Lineare Optimierung (MD3)
- Computational Statistics (MD6)
- Mathematische Logik (ME3)

Aus dem Master Mathematik sind aus den genannten Modulen die folgenden Veranstaltungen anrechenbar:

- Grundmodul Numerik und Optimierung (MM15):

Finite Elemente

Nichtlineare Optimierung

Numerische Optimierung bei Differentialgleichungen I

Uncertainty Quantification 1

- Grundmodul Statistik und Wahrscheinlichkeitsrechnung:

Wahrscheinlichkeitstheorie II

Statistik II

- Aufbaumodul Numerik und Optimierung (MM25):

Gemischte Finite Elemente

Parallele Löser für Finite Elemente

Numerische Optimierung bei Differentialgleichungen II

Uncertainty Quantification 2

- Spezialisierungsmodul Numerik und Optimierung (MM35):

Computational Fluid Dynamics

Fundamentals of Computational Environmental Physics

Mathematical Methods of Image and Pattern Analysis II

Konvexe Optimierung

Numerical Methods für Bayesian Inverse Problems

Numerical Methods for Maxwell's Equations

- Spezialisierungsmodul Statistik und Wahrscheinlichkeitsrechnung (MM36):

Statistische Analyse von maschinellen Lernalgorithmen

Aus den Ergänzungsmodulen:

- Berechenbarkeit und Komplexität I
- Berechenbarkeit und Komplexität II

2.5 Module aus dem M.Sc. Technische Informatik

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.