

Modulhandbuch
Masterstudiengang
„Angewandte Informatik“

Ruprecht-Karls-Universität Heidelberg
Fakultät für Mathematik und Informatik

Fassung vom 07.02.2024 zur Prüfungsordnung vom 22.07.2010
mit Änderung vom 07.02.2013
Rückwirkend für das Wintersemester 2023/24

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

Inhaltsverzeichnis

1	Qualifikationsziele, Profil und Besonderheiten des Masterstudiengangs Angewandte Informatik	4
1.1	Präambel - Qualifikationsziele der Universität Heidelberg	4
1.2	Profil des Studiengangs	4
1.3	Fachliche Qualifikationsziele des Studiengangs	4
1.4	Überfachliche Qualifikationsziele des Studiengangs	5
1.5	Erläuterungen zum Studiengang und den Modulbeschreibungen	6
1.5.1	Begründung für Module mit weniger als 5 LP	6
1.5.2	Beschreibung der Lehr- und Lernformen	6
1.5.3	Prüfungsmodalitäten	7
2	Studienverlaufsplan und Mobilität	8
2.1	Studienverlaufsplan	8
2.2	Mobilitätsfenster	8
3	Pflichtbereich	9
	Wissenschaftliches Arbeiten	10
	Seminar	12
	Masterarbeit	13
	Anwendungsgebiet	14
4	Wahlpflichtbereich	15
4.1	Gebietszuordnung der Module	15
4.2	Vertiefungen	17
	Bildverarbeitung	17
	Computergraphik und Visualisierung	18
	Information Systems Engineering	19
	Theoretische Informatik	21
	Wissenschaftliches Rechnen	22
4.3	Module aus der Informatik	23
	Advanced Machine Learning	24
	Algorithm Engineering	25
	Algorithms and Data Structures 2	27
	Artificial Intelligence for Programming	29
	Complex Network Analysis	31
	Computational Geometry	33
	Computer Graphics	34
	Computer Vision: 3D Reconstruction	35
	Computer Games (Game Engine Design)	36
	Discrete Structures 1	37
	Discrete Structures 2	38
	Distributed and Parallel Algorithms	39

Fortgeschrittenenpraktikum	41
Fundamentals of Machine Learning	42
Generative Neural Networks for the Sciences	43
Geometric Modeling and Animation	44
Hardware Aware Scientific Computing	46
Inverse Problems	48
IT Project Management	49
IT-Sicherheit 2	50
Machine Learning	52
Mining Massive Datasets	54
Natural Language Processing with Transformers	56
Object-Oriented Programming for Scientific Computing	58
Optimization for Machine Learning	59
Practical Geometry	61
Projektseminar Biomedizinische Bildanalyse	62
Randomisierte Algorithmen	64
Requirements Engineering	66
Scientific Visualization	67
Software Economics	68
Software Evolution	70
Visualisierung im Bereich Cultural Heritage	72
Volume Visualization	73
Knowledge Management and Decision-Making in Software Engineering	74
4.4 Module aus dem B.Sc./M.Sc. Mathematik	75
4.5 Module aus dem M.Sc. Physik	76
4.6 Module aus dem M.Sc. Technische Informatik	77

1 Qualifikationsziele, Profil und Besonderheiten des Masterstudiengangs Angewandte Informatik

1.1 Präambel - Qualifikationsziele der Universität Heidelberg

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 den Curricula und Modulen der einzelnen Studiengänge 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.

1.2 Profil des Studiengangs

Der Masterstudiengang Angewandte Informatik wird von der Fakultät für Mathematik und Informatik getragen. Der Masterstudiengang 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 und Absolventen sind qualifiziert für eigenverantwortliche und leitende Tätigkeiten. Der Masterstudiengang 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.

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

1.3 Fachliche Qualifikationsziele des Studiengangs

In fachlicher Hinsicht beherrschen die Absolventinnen und Absolventen des Masterstudiengangs Angewandte Informatik insbesondere die Kompetenzen des Bachelorstudiengangs, 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.

Die Absolventinnen und Absolventen des Masterstudiengangs Angewandte Informatik beherrschen 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.

1.4 Überfachliche Qualifikationsziele des Studiengangs

Die Absolventinnen und Absolventen des Masterstudiengangs 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.

1.5 Erläuterungen zum Studiengang und den Modulbeschreibungen

1.5.1 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.

1.5.2 Beschreibung der Lehr- und Lernformen

Vorlesung: Präsentation des Lehrstoffs durch die Lehrperson 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

1.5.3 Prüfungsmodalitäten

Zu Beginn jeder Veranstaltung werden die Details und insbesondere Abweichungen zu den unten aufgeführten Prüfungsmodalitäten von der Lehrperson mündlich und schriftlich bekannt gegeben.

Viele Module haben eine einheitliche Regelung bei der Vergabe der LP, daher wird diese Regelung hier einmal ausführlich beschrieben und bei den Modulbeschreibungen dann nur hierher verwiesen.

Regelung zur Vergabe der LP: In diesem Modul werden die LP bei bestandener Abschlussprüfung vergeben. Die Details zur Abschlussprüfung stehen bei den einzelnen Modulen. In diesem Modul gibt es einen Übungsbetrieb mit der Bearbeitung von Übungsaufgaben. Um zur Abschlussprüfung zugelassen zu werden, müssen mindest. 50% der Punkte in den Übungsaufgaben erreicht werden. Diese Zulassung gilt für das aktuelle und die beiden kommenden Semester (jeweils beide Prüfungszeiträume, siehe unten), d.h. bei jährlich angebotenen Modulen kann nach erfolgter Zulassung die Abschlussprüfung in diesem Semester oder ein Jahr später in den beiden Prüfungszeiträumen absolviert werden. Danach ist eine erneute Zulassung zur Abschlussprüfung im Übungsbetrieb zu erarbeiten.

Prüfungsschema: In diesem Feld der Modulbeschreibung ist eingetragen, wieviele Versuche zum Bestehen des Moduls laut Prüfungsordnung vorgesehen sind. Eine bestandene Prüfung kann nicht wiederholt werden.

1+1 besagt: dass nach dem ersten Versuch nur eine Wiederholungsmöglichkeit besteht.

Prüfungszeitraum: Für die schriftlichen Prüfungen (Klausuren) zum Ende jeden Semesters wurden zwei Prüfungszeiträume festgelegt. Der erste Prüfungszeitraum umfasst drei Wochen und besteht aus der letzten Woche der Vorlesungszeit und den ersten beiden Wochen der vorlesungsfreien Zeit. Der zweite Prüfungszeitraum umfasst vier Wochen und besteht aus den letzten drei Wochen der vorlesungsfreien Zeit und der ersten Woche der Vorlesungszeit. In Ausnahmefällen können Prüfungen außerhalb dieser Prüfungszeiträume stattfinden.

Prüfungstermine: Bei Modulen die einmal jährlich oder seltener angeboten werden, werden im Anschluss an das Modul immer zwei Prüfungstermine angeboten. Bei schriftlichen Prüfungen liegen diese innerhalb der oben genannten Prüfungszeiträume. Bei mündlichen Prüfungen werden die Termine von den Lehrenden festgelegt.

Bei Modulen, die in jedem Semester angeboten werden, gibt es im Anschluss an das Modul nur einen Prüfungstermin.

Die Studierenden wählen selbst, welche der angebotenen Prüfungstermine sie wahrnehmen.

Falls es Ausnahmen zu den Prüfungsterminen gibt, insbesondere wenn diese außerhalb der oben genannten Prüfungszeiträume liegen, müssen diese von der Lehrperson zu Beginn der Veranstaltung mündlich und schriftlich bekannt gegeben werden.

2 Studienverlaufsplan und Mobilität

2.1 Studienverlaufsplan

1. Jahr:	
Wissenschaftliches Arbeiten	2 LP
Seminar	4 LP
Anwendungsgebiet	6 LP
Wahlpflicht	48 LP
Summe	60 LP
2. Jahr:	
Seminar	4 LP
Anwendungsgebiet	12 LP
Wahlpflicht	14 LP
Masterarbeit	30 LP
Summe	60 LP
Gesamt:	120 LP

2.2 Mobilitätsfenster

Das Mobilitätsfenster für den Masterstudiengang Angewandte Informatik liegt in der Regel im zweiten und dritten Fachsemester, aber auch in den anderen Semestern kann ein Studienaufenthalt an einer anderen Hochschule im In- und Ausland durchgeführt werden. Im Master gibt es nur wenige Pflichtmodule, bei Modulen aus dem Wahlpflichtbereich oder dem Anwendungsgebiet ist eine Anerkennung durch die Wahlmöglichkeiten tendenziell einfacher.

Die Planungen für einen solchen Studienaufenthalt sollten frühzeitig begonnen werden, gerade für einen Auslandsaufenthalt kann diese Organisationsphase durchaus ein Jahr betragen.

Informationen zum Auslandsstudium finden Sie auf den Seiten des Erasmus Programms der Informatik <https://www.informatik.uni-heidelberg.de/erasmus>.

3 Pflichtbereich

Im Masterstudiengang 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	
LP 2	Dauer ein Semester	Angebotsturnus jedes Wintersemester
Format 2 SWS	Arbeitsaufwand 60 h; davon 30 h Präsenzstudium 30 h Selbststudium und praktische Übungen (eventuell in Gruppen)	Verwendbarkeit M.Sc. Angewandte Informatik
Sprache Deutsch oder Englisch	Lehrende wechselnd	Prüfungsschema 1+1
Lernziele	<p>Die Studierenden 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;</p> <p>kennen die Anforderungen an die Struktur von Anträgen zur Forschungsförderung.</p>	
Lerninhalte	<p>Literaturrecherche und -verwaltung Wissenschaftliches Vortragen, Schreiben, Publizieren und Begutachten Forschungsförderung über Drittmittel Forschungsmethoden und aktuelle Forschungsprojekte Wissenschaftliches Arbeiten nach dem Studium</p>	
Teilnahme- voraus- setzungen	keine	
Vergabe der LP und Modulendnote	<p>Das Modul wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Die Modulendnote wird durch die Note der Prüfung festgelegt. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.</p>	

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

Seminar

Code IS	Name Seminar	
LP 4	Dauer ein Semester	Angebotsturnus jedes Semester
Format 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
Sprache Deutsch oder Englisch	Lehrende je nach Angebot	Prüfungsschema 1+1
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>	
Lerninhalte	<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>	
Teilnahmevoraussetzungen	empfohlen sind: Kenntnisse im Themengebiet des Seminars	
Vergabe der LP und Modulendnote	<p>Das Modul wird mit einer benoteten Prüfung abgeschlossen. Diese Prüfung umfasst die Ausarbeitung und das Halten eines Vortrages von etwa 60 Minuten Dauer (inklusive Diskussion) sowie eine schriftliche Ausarbeitung von ca. 10 Seiten. Nähere Regelungen bezüglich des Formats der Ausarbeitung sowie der Präsentation werden zu Beginn der Lehrveranstaltung bekannt gegeben. Zur Vergabe der LP muss die Prüfung bestanden werden. Die Modulendnote wird durch die Note der Prüfung festgelegt.</p>	
Nuetzliche Literatur		

Masterarbeit

Code IMa	Name Masterarbeit	
LP 30	Dauer 6 Monate	Angebotsturnus jedes Semester
Format 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
Sprache Deutsch oder Englisch	Lehrende je nach Angebot	Prüfungsschema 1+1
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	
Lerninhalte	selbstständiges wissenschaftliches Bearbeiten einer beschränkten Aufgabenstellung aus der Informatik und ihren Anwendungen wissenschaftlicher Vortrag über die Problemstellung und die erarbeitete Lösung	
Teilnahme- voraus- setzungen	nach Prüfungsordnung mindestens 45 LP; weiterhin sind empfohlen: Wahlpflichtvorlesungen und Module Seminar (IS) und Fortgeschrittenenpraktikum (IFP)	
Vergabe der LP und Modulendnote	Zur Vergabe der LP ist das Bestehen der benoteten Masterarbeit nötig. Die Masterarbeit umfasst regelmäßige Treffen mit der Betreuerin bzw. dem Betreuer, die schriftliche Ausarbeitung und die Präsentation.	
Nuetzliche Literatur	wird von der Betreuerin bzw. dem Betreuer bekannt gegeben	

Anwendungsgebiet

Code IAG	Name Anwendungsgebiet	
LP 18	Dauer	Angebotsturnus
Format Vorlesung, Übung und / oder Praktikum	Arbeitsaufwand 540 h, Aufteilung in Präsenz- / Übungs- und Praktikumszeit in Absprache mit den Lehrenden	Verwendbarkeit M.Sc. Angewandte Informatik
Sprache Deutsch oder Englisch	Lehrende	Prüfungsschema
Lernziele	Vertiefte Kenntnisse und Fähigkeiten in einem Anwendungsgebiet	
Lerninhalte	<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>	
Teilnahmevoraussetzungen	empfohlen ist das gleiche Anwendungsgebiet wie im Bachelor	
Vergabe der LP und Modulendnote	<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.</p> <p>Die Module im IAG müssen benotet sein, unbenotete Module werden nur in begründeten Ausnahmefällen zugelassen.</p>	
Nuetzliche Literatur		

4 Wahlpflichtbereich

Im Wahlpflichtbereich sind insgesamt 62 LP zu erbringen. 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.

4.1 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

Advanced Machine Learning (IAML)
Artificial Intelligence for Programming (IAIP)
Fundamentals of Machine Learning (IFML)
Machine Learning (IML)

Computergraphik und Visualisierung (CGV)

Computational Geometry (ICGeo)
Computer Graphics (ICG)
Computer Games (Game Engine Design) (ICS)
Computer Vision: 3D Reconstruction (ICV3DR)
Geometric Modeling and Animation (IGMA)
Inverse Probleme (IIP)

Practical Geometrie (IPGeo)
Scientific Visualization (ISV)
Visualisierung im Bereich Cultural Heritage (IVCH)
Volume Visualization (IVV)

Datenbanksysteme (DB)

Complex Network Analysis (ICNA)

Optimierung

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

Parallele und Verteilte Systeme

Distributed and Parallel Algorithms (IDPA)
Mining Massive Datasets (IMMD)

Software Engineering (SWE)

IT-Projektmanagement (IPM)
Requirements Engineering (ISWRE)
Software Economics (ISWEco)
Software Evolution (ISWEvol)
Knowledge Management and Decision-Making in 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)
Randomisierte Algorithmen (IRA)

Wissenschaftliches Rechnen

Advanced Machine Learning (IAML)
Fundamentals of Machine Learning (IFML)
Generative Neural Networks for the Sciences (IGNNS)
Hardware Aware Scientific Computing (IHASC)
Machine Learning (IML)
Natural Language Processing with Transformers (INLPT)
Numerik (MD1)
Object-Oriented Programming for Scientific Computing (IOPSC)

4.2 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:

- Computational Geometry (ICGeo) 8 LP
- Computer Graphics (ICG) 8 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
- 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

Ergänzende Module:

- Artificial Intelligence for Programming (IAIP) 6 LP
- Fundamentals of Machine Learning (IFML) 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

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
Seminar - Lehrgebiet SWE	1 - 3	4
Seminar - Lehrgebiet DMA	1 - 3	4
ISE-Projekt	1 - 3	16
Vertiefende Module Lehrgebiet SWE	1 - 3	16
Vertiefendes Modul Lehrgebiet DMA	1 - 3	8
Weitere Module aus SWE oder DMA	1 - 3	14
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

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.

4.3 Module aus der Informatik

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

Advanced Machine Learning

Code IAML	Name Advanced Machine Learning	
LP 8	Dauer one semester	Angebotsturnus follows *Fundamentals of Machine Learning*
Format Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand 240h, thereof 60h lecture 90h tutorials, homework, lecture wrap-up 90h graded final report	Verwendbarkeit cannot be combined with *Machine Learning* M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Ullrich Köthe	Prüfungsschema 1+1
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.	
Lerninhalte	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)	
Teilnahmevoraussetzungen	recommended are: lecture *Fundamentals of Machine Learning* or similar	
Vergabe der LP und Modulendnote	The module is completed with a graded written examination. This examination is a report on a 90 h mini-research project. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations. Details will be given by the lecturer.	
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	
LP 8	Dauer one semester	Angebotsturnus every summer semester
Format Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand 240h; thereof 90h lectures and tutorials, 15h exam preparations, 135h lecture wrap-up and homework	Verwendbarkeit M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Christian Schulz	Prüfungsschema 1+1
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>	
Lerninhalte	<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>	
Teilnahme- voraus- setzungen	<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 (IADS2)</p>	
Vergabe der LP und Modulendnote	<p>The module is completed with a graded oral examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.</p>	

Nuetzliche Literatur	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 Jon M. Kleinberg, Éva Tardos: Algorithm design. Addison-Wesley 2006, ISBN 978-0-321-37291-8, pp. I-XXIII, 1-838 Stefan Näher: LEDA, a Platform for Combinatorial and Geometric Computing. Handbook of Data Structures and Applications 2004
---------------------------------	--

Algorithms and Data Structures 2

Code IADS2	Name Algorithms and Data Structures 2	
LP 8	Dauer one semester	Angebotsturnus every winter semester
Format 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 B.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. Scientific Computing
Sprache English	Lehrende Christian Schulz	Prüfungsschema 1+1
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 	
Lerninhalte	<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 	
Teilnahme-voraus-setzungen	<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>	
Vergabe der LP und Modulendnote	<p>The module is completed with a graded oral examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.</p>	

Nuetzliche Literatur	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 Kurt Mehlhorn, Peter Sanders: Algorithms and Data Structures: The Basic Toolbox. Springer 2008, ISBN 978-3-540-77977-3 Jon M. Kleinberg, Éva Tardos: Algorithm design. Addison-Wesley 2006, ISBN 978-0-321-37291-8, pp. I-XXIII, 1-838 Stefan Näher: LEDA, a Platform for Combinatorial and Geometric Computing. Handbook of Data Structures and Applications 2004
---------------------------------	---

Artificial Intelligence for Programming

Code IAIP	Name Artificial Intelligence for Programming	
LP 6	Dauer one semester	Angebotsturnus at least every 4th semester
Format 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 M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Artur Andrzejak	Prüfungsschema 1+1
Lernziele	Expected learning outcomes are: <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. 	
Lerninhalte	This module covers the following topics: <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. 	
Teilnahmevoraussetzungen	Skills in programming (preferably Python) and elementary knowledge of probability theory / statistics. Recommended prerequisites are lectures in machine learning, e.g. Foundations of machine learning.	
Vergabe der LP und Modulendnote	The module is completed with a graded oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	

Nuetzliche Literatur	Stuart J. Russell: Artificial intelligence: a modern ap-proach, (3rd ed.), Pearson, 2016, Heidi: https://bit.ly/2V9LQT9 Noah D. Goodman, Joshua B. Tenenbaum: Probabil-istic 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
---------------------------------	--

Complex Network Analysis

Code ICNA	Name Complex Network Analysis	
LP 8	Dauer one semester	Angebotsturnus every 2nd wintersemester
Format 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. Data and Computer Science M.Sc. Scientific Computing B.Sc. Mathematik
Sprache English	Lehrende Michael Gertz	Prüfungsschema 1+1
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 	
Lerninhalte	<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 	
Teilnahmevoraussetzungen	recommended are: Algorithmen und Datenstrukturen (IAD), Knowledge Discovery in Databases (IKDD), Lineare Algebra I (MA4)	

Vergabe der LP und Modulendnote	<p>The module is completed with a graded written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.</p>
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	
LP 8	Dauer one semester	Angebotsturnus irregular
Format 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 groups)	Verwendbarkeit M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Susanne Krömker	Prüfungsschema 1+1
Lernziele	<p>The students know the algorithms and data structures of geometric and topological data processing.</p> <p>They can understand and implement sweep algorithms for nearest neighbors, intersections of line segments and Voronoi diagrams, can construct alpha shapes and beta skeletons from pointclouds, know template-based and data-driven algorithms for the determination of isolines and isosurfaces, can work with discrete vector fields on simplicial complexes and know about persistence of topological invariants.</p> <p>They master the associated data structures for efficient storage and further processing and can calculate the complexity of the various algorithms.</p>	
Lerninhalte	Basic concepts from geometry, graph theory and topology, sweep algorithms in visibility analysis and Voronoi diagrams, Delaunay triangulations, alpha shapes, beta skeletons, isosurfaces, discrete Morse theory	
Teilnahmevoraussetzungen	recommended is: Algorithmen und Datenstrukturen (IAD)	
Vergabe der LP und Modulendnote	The module is completed with a graded oral examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
Nuetzliche Literatur	<p>Rolf Klein: Algorithmische Geometrie, Springer Verlag, 2005</p> <p>Herbert Edelsbrunner: Geometry and Topology of Mesh Generation, Cambridge University Press, 2001</p> <p>Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars: Computational Geometry - Algorithms and Applications, 3rd edition, Springer, 2008</p> <p>current publications</p>	

Computer Graphics

Code ICG	Name Computer Graphics	
LP 8	Dauer one semester	Angebotsturnus every 3rd semester
Format Lecture 4 SWS + Exercise course 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 B.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. Scientific Computing
Sprache English	Lehrende Filip Sadlo	Prüfungsschema 1+1
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.	
Lerninhalte	<ul style="list-style-type: none"> - Introduction - Perception and Color - Raytracing - Transformations - Rasterization - OpenGL - Textures - Spatial Data Structures 	
Teilnahme- voraus- setzungen	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
Vergabe der LP und Modulendnote	The module is completed with a graded written or oral examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
Nuetzliche Literatur	<p>P. Shirley, S. Marschner: Fundamentals of Computer Graphics, 3rd Edition, AK Peters</p> <p>OpenGL Specifications(GL 4.5 + GLSL 4.50) http://www.opengl.org/registry/ Optional</p> <p>A. S. Glassner: An Introduction to Ray Tracing, Academic Press</p> <p>T. Akenine-Möller, E. Haines: Real-Time Rendering, AK Peters, 2008</p>	

Computer Vision: 3D Reconstruction

Code ICV3DR	Name Computer Vision: 3D Reconstruction	
LP 6	Dauer one semester	Angebotsturnus every winter semester
Format 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 M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Carsten Rother	Prüfungsschema 1+1
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. 	
Lerninhalte	<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>	
Teilnahmevoraussetzungen	<p>recommended is a basic machine learning background (e.g. Fundamentals of Machine Learning, Advanced Machine Learning or equivalent)</p>	
Vergabe der LP und Modulendnote	<p>The module is completed with a graded examination. This examination is either a graded final report (about 10 pages) or an oral examination. The grade of this examination gives the grade for this module. Details for this examination as well as the requirements for the assignment of credits will be given by the lecturer at the beginning of this course.</p>	
Nuetzliche Literatur		

Computer Games (Game Engine Design)

Code ICS	Name Computer Games (Game Engine Design)	
LP 8	Dauer one semester	Angebotsturnus every summer semester
Format Lecture 3 SWS + Exercise 3 SWS	Arbeitsaufwand 240 h; thereof 75 h lectures and tutorials 15 h exam preparations 150 h self-study and exercises	Verwendbarkeit M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Jürgen Hesser	Prüfungsschema 1+1
Lernziele	The students understand the game engine concepts, the decision for specialized class structures, support tools, and the typical architectural elements and are able to apply these concepts in developing an own game engine. They are able to apply and further develop methods for visualizing 3D scenes, perform collision detection and hereby are able to identify the appropriate algorithms. They have the capability to develop animation methods with different levels of complexity and are able to assess which method to take under the trade-off between performance and quality. Students will be able to find and apply appropriate techniques for path planning, to improve the found paths to be more realistic. Finally, they are able to identify the different concepts of AI in games and develop and apply these techniques for own games. In the exercises, they apply the theoretical concepts and program applications in order to see how to translate concepts into code.	
Lerninhalte	Overview of the structure and the components of computer games Architecture of Game Engines Elements of the Graphics Subengine Algorithms for Collision Detection Animation techniques and physics Path planning and AI	
Teilnahme- voraus- setzungen	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
Vergabe der LP und Modulendnote	The module is completed with a graded oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
Nuetzliche Literatur	Gregory et al: Game Engine Architecture Ericson: Real-Time Collision Detection Eberly: Game Physics Millington: Artificial Intelligence for Games	

Discrete Structures 1

Code IDS1	Name Discrete Structures 1	
LP 8	Dauer one semester	Angebotsturnus every winter semester
Format 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 B.Sc. Informatik B.Sc. Mathematik
Sprache English	Lehrende Felix Joos	Prüfungsschema 1+1 (im BSc Mathematik 1+2)
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 	
Lerninhalte	<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 	
Teilnahme- voraus- setzungen	recommended are: Einführung in die Praktische Informatik (IPI), Mathematik für Informatiker 1 (IMI1) or Lineare Algebra 1 (MA4), Mathematik für Informatiker 2 (IMI2) or Analysis 1 (MA1)	
Vergabe der LP und Modulendnote	The module is completed with a graded oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
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	
LP 8	Dauer one semester	Angebotsturnus irregularly in the summer semester
Format 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. Data and Computer Science M.Sc. Mathematik
Sprache English	Lehrende Felix Joos	Prüfungsschema 1+1
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 	
Lerninhalte	<ul style="list-style-type: none"> - Probabilistic Methods - Extremal graph theory - Expander graphs - Quasirandom graphs - Further advanced topics 	
Teilnahmevoraussetzungen	recommended is: Discrete Structures 1	
Vergabe der LP und Modulendnote	The module is completed with a graded oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
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. 	

Distributed and Parallel Algorithms

Code IDPA	Name Distributed and Parallel Algorithms	
LP 8	Dauer one semester	Angebotsturnus every 3rd to 4th semester
Format 4 SWS lecture 2 SWS tutorial, homework assignments	Arbeitsaufwand 240h; thereof 90h lectures and tutorials, 15h exam preparations, 135h lecture wrap-up and homework	Verwendbarkeit M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Christian Schulz	Prüfungsschema 1+1
Lernziele	Students understand fundamental theoretical and practical concepts of advanced parallel 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 where parallel or distributed algorithms are used, are able to analyse new distributed and parallel algorithms as well as analysing their running time, and select appropriate algorithms for parallel and distributed applications ? students are able to apply parallel and distributed algorithms and data structures to real-world problems, and can objectively assess the quality of the results.	
Lerninhalte	Introduction to distributed and parallel algorithms, PRAM model, design and analysis of parallel and distributed algorithms, isoefficiency, UMA vs. NUMA, memory consistency for shared-memory, communication models (with and without network, fully interconnected with half duplex or full duplex, BSP), critical path lengths, parallel associative operations, reduction operations, matrix multiplication, broadcast operations, MPI basic toolbox, ranking, parallel sorting (multiway merge, quick sort, sample sort), prefix sums, all-to-all communication, map-reduce, list ranking, parallel graph algorithms (minimum spanning trees, connected components, shortest paths, graph partitioning), process mapping, communication-free parallel graph generation, parallel sampling algorithms.	
Teilnahme- voraus- setzungen	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Lineare Algebra 1	
Vergabe der LP und Modulendnote	The module is completed with a graded oral examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	

Nuetzliche Literatur	<ul style="list-style-type: none">- Sanders, Mehlhorn, Dietzfelbringer, Dementiev. Sequential and Parallel Algorithms and Data Structures. 2019.- Kumar, Grama, Gupta, Karypis. Introduction to Parallel Computing. Design and Analysis of Algorithms. 1994- Leighton. Introduction to Parallel Algorithms and Architectures. 1992- Jaja. An Introduction to Parallel Algorithms. 1992
---------------------------------	---

Fortgeschrittenenpraktikum

Code IFP	Name Fortgeschrittenenpraktikum	
LP 8	Dauer	Angebotsturnus jedes Semester
Format Praktikum 6 SWS	Arbeitsaufwand 240 h; davon mind. 25 h Präsenzzeit 10 h Vorbereitung Vortrag	Verwendbarkeit B.Sc. Angewandte Informatik B.Sc. Informatik Lehramt Informatik
Sprache Deutsch oder Englisch	Lehrende je nach Angebot	Prüfungsschema 1+1
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): Durchführung und Evaluation von Projekten und ihrer Phasenstruktur 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>	
Lerninhalte	<p>Domänenkenntnisse abhängig von den Lehrenden; allgemeine Lerninhalte sind: Vertiefung in die Projektarbeit Eigenständige Entwicklung von komplexer Software und deren Dokumentation</p>	
Teilnahmevoraussetzungen	empfohlen sind: Anfängerpraktikum (IAP), Einführung in Software Engineering (ISW)	
Vergabe der LP und Modulendnote	Das Modul wird mit einer benoteten Prüfung abgeschlossen. Diese Prüfung umfasst die Bewertung der dokumentierten Software, des Projektberichts (5-10 Seiten) und des Vortrags (ca. 30 Minuten zzgl. Diskussion). Zur Vergabe der LP muss diese Prüfung bestanden werden. Die Modulendnote wird durch die Note der Prüfung festgelegt.	
Nuetzliche Literatur		

Fundamentals of Machine Learning

Code IFML	Name Fundamentals of Machine Learning	
LP 8	Dauer one semester	Angebotsturnus in (irregular) alternation with *Machine Learning*
Format Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand 240h, thereof 60h lecture 90h tutorials, homework, lecture wrap-up 90h graded final report	Verwendbarkeit cannot be combined with *Machine Learning* M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Ullrich Köthe	Prüfungsschema 1+1
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.	
Lerninhalte	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)	
Teilnahmevoraussetzungen	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
Vergabe der LP und Modulendnote	The module is completed with a graded written examination. This examination is a report on a 90 h mini-research project. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations. Details will be given by the lecturer.	
Nuetzliche Literatur	Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning (2nd edition), Springer, 2009	

Generative Neural Networks for the Sciences

Code IGNNS	Name Generative Neural Networks for the Sciences	
LP 8	Dauer one semester	Angebotsturnus in (irregular) alternation with *Machine Learning*
Format Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand 240h, thereof 60h lecture 90h tutorials, homework, lecture wrap-up 90h graded final report	Verwendbarkeit M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Ullrich Köthe	Prüfungsschema 1+1
Lernziele	Participants get to know a broad range of generative neural network design and learning methods, with an emphasis on solving problems in the sciences. They understand the strengths and limitations of these methods, can apply them to real-world problems and objectively assess the quality of the results. In addition, students familiarize themselves with important open-source implementations of these methods.	
Lerninhalte	Types of generative neural networks: normalizing flows, diffusion models, (variational) autoencoders, recurrent networks, transformers Techniques: simulation-based inference, hierarchical models, physic-informed neural networks, symbolic regression, causal discovery Quality diagnostics: predictive accuracy, probabilistic calibration, re-simulation error, disentanglement scores, generalization ability, and pitfalls of those diagnostics Applications: design of efficient surrogates for classical models, Bayesian inference for inverse problems, analysis of dynamic systems, with examples from physics, medicine, engineering, cognitive science, and others	
Teilnahmevoraussetzungen	recommended: basic knowledge of deep learning and statistics	
Vergabe der LP und Modulendnote	The module is completed with a graded written exam. This exam is a report on a 90 h mini-research project. The final grade of the module is determined by the grade of the exam. The requirements for the assignment of credits follows the regulations in section modalities for exams. Details will be given by the lecturer.	
Nuetzliche Literatur	Kevin Murphy. Probabilistic Machine Learning: Advanced Topics (2023)	

Geometric Modeling and Animation

Code IGMA	Name Geometric Modeling and Animation	
LP 8	Dauer one semester	Angebotsturnus every 3rd semester
Format 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 M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Filip Sadlo	Prüfungsschema 1+1
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>	
Lerninhalte	<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>	
Teilnahme- voraus- setzungen	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
Vergabe der LP und Modulendnote	The module is completed with a graded oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	

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	
LP 8	Dauer one semester	Angebotsturnus irregular
Format 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 M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Peter Bastian	Prüfungsschema 1+1
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.	
Lerninhalte	<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 	
Teilnahmevoraussetzungen	basic knowledge in computer architecture and numerical methods; good programming skills in C++	
Vergabe der LP und Modulendnote	The module is completed with a graded examination. The final grade of the module is determined by the grade of the examination. Details for this examination as well as the requirements for the assignment of credits will be given by the lecturer at the beginning of this course.	

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	
LP 8	Dauer one semester	Angebotsturnus every summer semester
Format Lecture 2 SWS + Exercise course 2 SWS + Homework	Arbeitsaufwand 240 h; thereof 60 h lectures and tutorials 15 h exam preparations 165 h self-study and exercises / homework	Verwendbarkeit M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Sprache English	Lehrende Jürgen Hesser	Prüfungsschema 1+1
Lernziele	Students understand the mathematical properties of inverse problems and are able to demonstrate and show why these problems are difficult to solve. They will learn principles of how to solve both deterministic and stochastic problems, they are able to identify problem settings which request specific deterministic or stochastic approaches and the regularization methods therein. They are able to select an appropriate regularization parameter strategy and understand their differences in particular. They also understand how to formulate and apply compressed sensing and deep learning for inverse problems. All principles are presented in selected areas in parameter estimation. Students will gain the competence in solving complex problems that cannot be dealt with classical techniques and they will be able to adequately evaluate complex experimental measurements.	
Lerninhalte	Definition of ill-posedness Deterministic inverse problems, regularization techniques Tikhonov regularization, data and model resolution matrix, pseudo-inverses Stochastic inverse problems and Bayes theorem Regularization parameter selection Compressed sensing Deep Learning for Inverse Problems	
Teilnahme- voraus- setzungen	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Numerische Mathematik	
Vergabe der LP und Modulendnote	The module is completed with a graded written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
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 Project Management

Code IPM	Name IT Project Management	
LP 3	Dauer one semester	Angebotsturnus every second winter semester
Format lecture + exercise 2 SWS	Arbeitsaufwand lecture + exercise 90 h all together, thereof: 30 h lecture + exercise 15 h preparation for exam 45 h self-study and homework (optionally in groups)	Verwendbarkeit M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Sprache English	Lehrende Andrea Herrmann	Prüfungsschema 1+1
Lernziele	The participants are able to plan and control a project, understand, how projects are embedded into organizations, and have basic knowledge about contractual questions.	
Lerninhalte	project planning, project organization cost estimation project offer, contract, negotiations process models risk management controlling IT contract laws change management time management project closure distributed software engineering	
Teilnahme- voraus- setzungen	none	
Vergabe der LP und Modulendnote	The module is completed with a graded (oral or written) examination. The grade of the module is the grade of the examination. Prerequisite for the participation in the exam are 50% of the points for the homework.	
Nuetzliche Literatur	PMI (Project Management Institute): A Guide to the Project Management Body of Knowledge (PM BOK ® Guide), 6th Edition, 2017	

IT-Sicherheit 2

Code IITS2	Name IT-Sicherheit 2	
LP 6	Dauer ein Semester	Angebotsturnus unregelmäßig
Format Vorlesung 2 SWS + Übung 2 SWS	Arbeitsaufwand 180 h; davon 60 h Präsenzstudium 15 h Prüfungsvorbereitung 105 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	Verwendbarkeit nicht kombinierbar mit Modul IT-Sicherheit für 8 LP M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Sprache Deutsch	Lehrende Vincent Heuveline	Prüfungsschema 1+1
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 erweiterte Kenntnisse über die Sicherung großer Netzwerke und der Kommunikationsinfrastruktur (Routing, Namensauflösung, Internet-Firewalls, Intrusion Detection Systeme). * erwerben vertiefte Kompetenzen zur Detektion von Cyberangriffen. * erwerben grundlegende Kompetenzen im Bereich Penetration Testing. <p>Langfristiges Ausbildungsziel: Einsatz- und Beschäftigungsfähigkeit in der Breite des Arbeitsfeldes IT-Sicherheit.</p>	
Lerninhalte	<p>Der IT-Sicherheit kommt bei der allgegenwärtigen Digitalisierung eine Schlüsselrolle zu. Die Vorlesung It-Sicherheit 2 vermittelt methodische Ansätze zur Modellierung und Bewertung von Angriffsszenarien, auf Basis welcher technische Gegenmaßnahmen umgesetzt werden können. Insbesondere werden folgende Schwerpunkte adressiert:</p> <ul style="list-style-type: none"> ?[9]9]Sicherheitsmodelle und Bewertungskriterien ?[9]9]Authentifikationsverfahren ?[9]9]Schutz von Kommunikationsinfrastruktur; Netzsicherheit ?[9]9]Digitale Identität ?[9]9]Software-Exploitation ?[9]9]Penetration Testing ?[9]9]Zero Trust Security <p>Mit Hilfe von virtuellen Maschinen in einem geschützten Bereich werden klassische Angriffs- und Schutzszenarien praktisch untersucht.</p>	
Teilnahmevoraussetzungen	empfohlen ist: IT-Sicherheit 1 (IITS1)	

Vergabe der LP und Modulendnote	Das Modul wird mit einer benoteten Klausur abgeschlossen. Die Modulendnote wird durch die Note der Klausur festgelegt. Für die Vergabe der LP gilt die Regelung aus dem Kapitel Prüfungsmodalitäten.
Nuetzliche Literatur	

Machine Learning

Code IML	Name Machine Learning	
LP 8	Dauer one semester	Angebotsturnus in (irregular) alternation with *Fundamentals of Machine Learning* + *Advanced Machine Learning*
Format Lecture 4 SWS + Exercise course 2 SWS	Arbeitsaufwand Arbeitsaufwand: 240h, thereof 60h lecture 90h tutorials, homework, lecture wrap-up 90h graded final report	Verwendbarkeit cannot be combined with *Fundamentals of Machine Learning* or *Advanced Machine Learning* M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Ullrich Köthe	Prüfungsschema 1+1
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.	
Lerninhalte	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	
Teilnahmevoraussetzungen	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
Vergabe der LP und Modulendnote	The module is completed with a graded written examination. This examination is a report on a 90 h mini-research project. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations. Details will be given by the lecturer.	

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	
LP 6	Dauer one semester	Angebotsturnus at least every 4th semester
Format 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 M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Artur Andrzejak	Prüfungsschema 1+1
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 	
Lerninhalte	<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 	
Teilnahme-voraus-setzungen	recommended are Knowledge of Java/Python and in elementary probability theory / statistics; module IBD can be taken as a complement / extension.	
Vergabe der LP und Modulendnote	The module is completed with a graded examination. The final grade of the module is determined by the grade of the examination. Details for this examination as well as the requirements for the assignment of credits will be given by the lecturer an the beginning of this course.	

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.mmms.org/) * Trevor Hastie, Robert Tibshirani, Jerome Friedman, 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)
-----------------------------	--

Natural Language Processing with Transformers

Code INLPT	Name Natural Language Processing with Transformers	
LP 6	Dauer one semester	Angebotsturnus every 2nd winter semester
Format Lecture 2 h + Exercise course 2 h	Arbeitsaufwand 180 h; thereof 60 h lecture 120 h self-study and working on assignments/projects (optionally in groups)	Verwendbarkeit M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Michael Gertz	Prüfungsschema 1+1
Lernziele	<p>Students</p> <ul style="list-style-type: none"> - fully understand the principles and methods underlying word embedding approaches - are familiar with traditional sequence-to-sequence machine learning methods - can describe the key concepts and techniques underlying attention mechanism and different transformer architectures - understanding training and fine-tuning approaches to improve the performance of different transformer architectures for different downstream NLP tasks - know the key methods and architectural components for building QA and text summarization pipelines - can build and deploy QA and text summarization pipelines using common software frameworks - know key metrics in evaluating transformer architectures for different applications - can implement diverse transformer-based NLP applications using common Python frameworks and libraries - can deploy transformer-based NLP applications through Web interfaces 	
Lerninhalte	<ul style="list-style-type: none"> - Word embeddings (review of simple neural network architectures and concepts) - Sequence-to-sequence models (Recurrent Neural Networks, LSTM, GRU) - Attention mechanism - Transformer components (encoder, decoder) and common transformer architectures (BERT, GPT, T5) - Training and fine-tuning transformers, including zero- and few-shot learning - Text summarization approaches - Question answering and building a QA pipeline - Transformer architectures for conversational AI - Programming and model frameworks such as Huggingface, LangChain, OpenAI and (cloud-based) vector databases 	

Teilnahmevoraussetzungen	<p>Recommended courses: Data Science for Text Analytics (IDSTA), Foundations of Machine Learning (IML)</p> <p>Recommended background: solid knowledge of basic calculus, statistics, and linear algebra; good Python programming skills; familiarity with frameworks such as Huggingface, Google Colab, and cloud-based services, in particular vector databases</p>
Vergabe der LP und Modulendnote	<p>Assignments (40%) and Programming Project (60%); about 4-5 assignments focusing on the material learned in class on a conceptual and formal level; group project in which 3-4 students develop a prototypical transformer-based application, 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. Both assignments and project must be at least satisfactory (4,0) in order to pass the class.</p>
Nuetzliche Literatur	<p>For the different topics, several research papers will be provided to students via the Moodle platform. The following textbooks are useful but not required.</p> <ul style="list-style-type: none"> - Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Natural Language Processing with Transformers, 2022 (revised edition) - Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft) <p>Furthermore, during the course of this lecture, several papers covering topics discussed in class will be provided.</p>

Object-Oriented Programming for Scientific Computing

Code IOPSC	Name Object-Oriented Programming for Scientific Computing	
LP 6	Dauer one semester	Angebotsturnus irregularly
Format 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 B.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. Scientific Computing
Sprache English	Lehrende varying	Prüfungsschema 1+1
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.	
Lerninhalte	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	
Teilnahmevoraussetzungen	recommended are: Einführung in die Praktische Informatik (IPI), alternatively basic knowledge of an object-oriented programming language	
Vergabe der LP und Modulendnote	The module is completed with a graded written examination. The final grade of the module is determined by the grade of the examination. The lecture will give the requirements for the assignment of credits.	
Nuetzliche Literatur		

Optimization for Machine Learning

Code IOML	Name Optimization for Machine Learning	
LP 8	Dauer one semester	Angebotsturnus every winter semester
Format 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 M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Mathematik M.Sc. Scientific Computing
Sprache English	Lehrende Bogdan Savchynskyy	Prüfungsschema 1+1
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 	
Lerninhalte	<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. 	
Teilnahmevoraussetzungen	recommended are: linear algebra, analysis and any universal programming language (e.g. C/C++/Pascal/python)	

Vergabe der LP und Modulendnote	The module is completed with a graded oral examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.
Nuetzliche Literatur	will beannounced by the lecturer at the beginning of the course

Practical Geometry

Code IPGeo	Name Practical Geometry	
LP 4	Dauer one semester	Angebotsturnus irregularly
Format Lecture 2 SWS + Exercise course 1 SWS	Arbeitsaufwand 120h, thereof 45 h lecture 60 h self-study and working on assignments 15h preparation for exam	Verwendbarkeit M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Susanne Krömker	Prüfungsschema 1+1
Lernziele	Understanding of basic geometric concepts for data analysis as well as efficient point search and further processing of measurement data Confident handling of projections and descriptions beyond the three-dimensional world of experience Calculation of geometric invariants, distances, curvatures from measurement data, reconstructed and generated surfaces	
Lerninhalte	Basic areas of geometry with relevance in computer graphics, image processing, pattern recognition, computer vision and geometric modeling. (i) Analytic geometry: operations on vector spaces with appropriate coordinates and mappings (affine mappings, collinearities), geometric fitting of point clouds to linear structures or planes from error-prone measurement data (ii) Projective geometry: central projection and inverse reconstruction of 3D objects from planar images (computer vision, geodesy), differences between B-spline curves and surfaces and the class of NURBS, freeform geometries in CAD systems (iii) Differential geometry: parameter representations in geometric data processing, implicit representations (level sets), estimation of invariants from discrete data (triangulations, point clouds).	
Teilnahmevoraussetzungen	recommended are: linear algebra, computational geometry and any programming language (e.g. C/C++/Pascal/python)	
Vergabe der LP und Modulendnote	The module is completed with a graded oral examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
Nuetzliche Literatur	Geometrie für Informatiker, Script TU Vienna 2004, Helmut Pottmann, current publications	

Projektseminar Biomedizinische Bildanalyse

Code IPBB	Name Projektseminar Biomedizinische Bildanalyse	
LP 6	Dauer ein Semester	Angebotsturnus jedes Sommersemester
Format 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 M.Sc. Data and Computer Science
Sprache Deutsch	Lehrende Karl Rohr	Prüfungsschema 1+1
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>	
Lerninhalte	<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>	
Teilnahmevoraussetzungen	empfohlen sind: Grundkenntnisse in Bildverarbeitung (Computer Vision, Image Analysis), Programmierkenntnisse, Kenntnisse in Software Engineering	

Vergabe der LP und Modulendnote	<p>Das Modul wird mit einer benoteten Prüfung abgeschlossen. Diese Prüfung umfasst Vortragspräsentationen von Zwischen- und Endergebnissen (jeder Studierende 4 Vorträge je ca. 10 Min. und anschließender Diskussion) und eine schriftliche Ausarbeitung der theoretischen Grundlagen, der verwendeten Methoden und der Ergebnisse (jeder Studierende ca. 10 Seiten). Zur Vergabe der LP muss diese Prüfung bestanden werden. Die Modulendnote wird durch die Note der Prüfung festgelegt.</p>
Nuetzliche Literatur	<p>Bekanntgabe in der Lehrveranstaltung</p>

Randomisierte Algorithmen

Code IRA	Name Randomisierte Algorithmen	
LP 6	Dauer ein Semester	Angebotsturnus mindst. jedes 4. Semester
Format Vorlesung 3 SW + Übung 1 SWS	Arbeitsaufwand 180 h; davon 60 h Präsenzstudium 40 h Prüfungsvorbereitung 80 h Selbststudium und Bearbeitung der Übungsaufgaben (eventuell in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. Scientific Computing
Sprache Deutsch	Lehrende Wolfgang Merkle	Prüfungsschema 1+1
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.	
Lerninhalte	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	
Teilnahmevoraussetzungen	empfohlen sind: elementare Grundkenntnisse in Algorithmen wie sie z.B. im Modul Algorithmen und Datenstrukturen (IAD) vermittelt werden.	
Vergabe der LP und Modulendnote	Das Modul wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Die Modulendnote wird durch die Note der Prüfung festgelegt. Für die Vergabe der LP gilt die Regelung aus dem Kapitel Prüfungsmodalitäten.	

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	
LP 8	Dauer ein Semester	Angebotsturnus jedes 2. Sommersemester
Format 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 B.Sc. Informatik M.Sc. Angewandte Informatik
Sprache Deutsch	Lehrende Barbara Paech	Prüfungsschema 1+1
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)	
Lerninhalte	Methoden, Prozesse und Werkzeuge für Unternehmensmodellierung Prozessverbesserung in Unternehmen Anforderungserhebung und -verhandlung Requirements Management (Verbreitung, Prüfung und Aktualisierung von Anforderungen)	
Teilnahme- voraus- setzungen	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden	
Vergabe der LP und Modulendnote	Das Modul wird mit einer benoteten Klausur abgeschlossen. Die Modulendnote wird durch die Note der Klausur festgelegt. Weitere Details zur Vergabe der LP werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.	
Nuetzliche Literatur	Wird jährlich aktualisiert	

Scientific Visualization

Code ISV	Name Scientific Visualization	
LP 8	Dauer one semester	Angebotsturnus every 3rd semester
Format 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 M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Sprache English	Lehrende Filip Sadlo	Prüfungsschema 1+1
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.	
Lerninhalte	<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 	
Teilnahme-voraus-setzungen	strongly recommended is: Computer Graphics (ICG) recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
Vergabe der LP und Modulendnote	The module is completed with a graded oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
Nuetzliche Literatur	C.D. Hansen, C.R. Johnson, The Visualization Handbook, 2005.	

Software Economics

Code ISWEco	Name Software Economics	
LP 3	Dauer one semester	Angebotsturnus irregularly
Format Lecture 2 SWS	Arbeitsaufwand 90 hours, thereof 30 hours lecture 35 hours individual processing / self-study 25 hours preparation for exam (in groups possible / recommended)	Verwendbarkeit M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Sprache English	Lehrende Eckhart von Hahn	Prüfungsschema 1+1
Lernziele	<p>After a successful participation in the lecture the student can</p> <ul style="list-style-type: none"> - roughly determine the price and licensing of an already created software, - plan and initiate marketing activities for software and software-related services / products, - roughly understand the balance sheet and profit-and-loss statement of a software manufacturer, - assess the value of a software with its various components, from the perspective of the manufacturer as well as from the perspective of the user, - plan price negotiations for software projects. <p>The student knows afterwards</p> <ul style="list-style-type: none"> - the basics of cost and performance accounting (as far as it is relevant for software creation), - the different types of (legal) contracts that are used in the area of software creation, - the most important negotiation strategies when negotiating software contracts, - legal aspects in the area of IT crime, - as well as the relevance of the lecture topics in the practice of industrial software creation. 	

Lerninhalte	<p>This module teaches those basic concepts of economics, which are relevant for software creation or software service delivery. The content of the lecture is assembled on the background of the lecturer's doctoral research and 20 years of corresponding Software Engineering experience in the (industrial) practice, based on current and classical literature:</p> <ul style="list-style-type: none"> - disambiguation of terms, - economic aspects during the planning and creation phase of the software lifecycle - economic aspects during the value assessment phase - economic aspects during the value transfer phase - accounting aspects - maintaining the value of software
Teilnahmevoraussetzungen	<p>Recommended are knowledge and skills taught in the module Introduction to Software Engineering (ISW)</p>
Vergabe der LP und Modulendnote	<p>The module is concluded with a graded exam - oral or written. Details are provided at the beginning of the lecture.</p>
Nuetzliche Literatur	<p>Boehm, B.W.: Software Engineering Economics. New Jersey 1981 Buxmann , P.; Diefenbach, H.; Hess, T.: Die Softwareindustrie. Ökonomische Prinzipien, Strategien, Perspektiven. Heidelberg, 2015 Versteegen, G.: Marketing in der IT Branche. Heidelberg 2003 von Hahn, E.: Werterhaltung von Software. Wiesbaden 2005 Wöhe, G.; Döring, U., Brösel, G.: Einführung in die Allgemeine Betriebswirtschaftslehre. München 2020</p>

Software Evolution

Code ISWEvol	Name Software Evolution	
LP 3	Dauer one semester	Angebotsturnus irregularly
Format Lecture 2 SWS	Arbeitsaufwand 90h, thereof 30h lecture 35h individual processing / self-study 25h preparation for exam (in groups possible / recommended)	Verwendbarkeit M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Sprache English	Lehrende Eckhart von Hahn	Prüfungsschema 1+1
Lernziele	<p>After the successful participation in the lecture the student can</p> <ul style="list-style-type: none"> - create a maintenance concept for an existing software, - plan a software reengineering project from a technical / functional perspective, <p>- develop a framework to enable a sustainable software development during the initial creation phase.</p> <p>The student knows afterwards</p> <ul style="list-style-type: none"> - the typology of software maintenance and the management of troubleshooting, - the classical array of software revitalization techniques (e.g. refactoring), - the difference and the challenges of progressive development versus the initial creation of software and on which aspects you have to pay particular attention – through the lense of the provider of a software as well as the user of a software, - in general the relevance of the topic for the industrial engineering practice. 	
Lerninhalte	<p>This module intends to convey the concepts for a successful software engineering lifecycle after its initial creation.</p> <p>The content of the lecture is assembled on the background of the lecturers doctoral research and 20 years of corresponding Software Engineering experience in the (industrial) practice, based on current and classical literature:</p> <ul style="list-style-type: none"> - Disambiguation of terms - Software Maintenance - Software Reengineering - Progressive Software Development / Software Evolution in particular and its management - Software Migration 	
Teilnahmevoraussetzungen	Recommended are knowledge and skills taught in the module Introduction to Software Engineering (ISW)	

Vergabe der LP und Modulendnote	The module is concluded with a graded exam – oral or written. Details are provided at the beginning of the lecture.
Nuetzliche Literatur	<p>Alt, R.; Auth, G.; Kögler, C.: Innovationsorientiertes IT-Management mit DevOps – IT im Zeitalter von Digitalisierung und Software-defined Business. Wiesbaden 2017.</p> <p>Arnold, R. (Hrsg.): Software Reengineering. Los Alamitos 1993.</p> <p>Fowler, M.: Refactoring – Improving the Design of Existing Code. Reading, Massachusetts, 1999.</p> <p>Furrer, F.J.: Future-Proof Software-Systems. Wiesbaden 2019.</p> <p>von Hahn, E.: Werterhaltung von Software. Wiesbaden 2005.</p> <p>Lilienthal, C.: Langlebige Software-Architekturen. Heidelberg, 2017.</p> <p>Müller, B.: Reengineering. Eine Einführung. Stuttgart 1997.</p> <p>Reussner, R.; Goedicke, M.; Hasselbring, W.; Vogel-Heuser, B.; Keim, J.; Martin, L. (Herausgeber): Managed Software Evolution. Cham 2019.</p> <p>Sneed, H.M.; Hasitschka, M.; Teichmann, M.-T.: Software-Produktmanagement. Wartung und Weiterentwicklung bestehender Anwendungssysteme. Heidelberg 2005.</p> <p>Smith, D.D.: Designing Maintainable Software. Heidelberg 1999.</p>

Visualisierung im Bereich Cultural Heritage

Code IVCH	Name Visualisierung im Bereich Cultural Heritage	
LP 2	Dauer ein Semester	Angebotsturnus unregelmäßig
Format Vorlesung 2 SWS	Arbeitsaufwand 60 h; davon 30 h Präsenzstudium, 30 h Prüfungsvorbereitung	Verwendbarkeit B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Angewandte Informatik
Sprache Deutsch	Lehrende Susanne Krömker	Prüfungsschema 1+1
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).	
Lerninhalte	Weißlicht- und Time-of-flight-Scanner, Rekonstruktionen von Gefäßen und Gebäuden, 3D-Puzzle, Skelettierung, ethische Grundsätze	
Teilnahmevoraussetzungen	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Computergraphik 1 (ICG1)	
Vergabe der LP und Modulendnote	Das Modul wird mit einer benoteten mündlichen Prüfung abgeschlossen. Die Modulendnote wird durch die Note der Prüfung festgelegt. Für die Vergabe der LP gilt die Regelung aus dem Kapitel Prüfungsmodalitäten.	
Nützliche Literatur	Clive Orton: Mathematics in Archaeology. Cambridge, MA, Cambridge University Press, 1982 Katsushi Ikeuchi, Daisuke Miyazaki (editors): Digitally Archiving Cultural Objects. Springer, 2007	

Volume Visualization

Code IVV	Name Volume Visualization	
LP 8	Dauer one semester	Angebotsturnus every summer semester
Format Lecture 4 SWS + Exercise course 3 SWS	Arbeitsaufwand 240 h; thereof 75 h lecture 15 h preparation for exam 150 h self-study and working on assignments/projects (optionally in groups)	Verwendbarkeit M.Sc. Angewandte Informatik, M.Sc. Data and Computer Science,
Sprache English	Lehrende Jürgen Hesser	Prüfungsschema 1+1
Lernziele	<p>The students learn to understand how to use techniques of volume visualization to render complex scientific data.</p> <p>This consists of the representation of data by surface or volume elements, the conversion of different representations and techniques of interpolation.</p> <p>They further understand the physical principles of volume rendering, the different strategies of their realization with advantages and disadvantages ? they should critically assess different techniques ? and their parallelization.</p>	
Lerninhalte	<p>Introduction of the visualization of scientific data of natural sciences and bio-sciences</p> <p>Discrete and continuous representation of data and methods of interpolation</p> <p>Methods of conversion between surface- and volumerepresentations and their efficient realizations</p> <p>Theory of volume rendering and their different realizations</p> <p>Accelerating and parallelization of volume rendering</p> <p>Programming technique: GPU-programming</p>	
Teilnahmevoraussetzungen	recommended are: Introduction into computer science I (IPI), programming course (IPK), algorithms & data structures (IAD);	
Vergabe der LP und Modulendnote	The module is completed with a graded written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
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	

Knowledge Management and Decision-Making in Software Engineering

Code ISWKM	Name Knowledge Management and Decision-Making in Software Engineering	
LP 3	Dauer one semester	Angebotsturnus every second winter semester
Format lecture + exercise 2 SWS	Arbeitsaufwand lecture + exercise 90 h all together, thereof: 30 h lecture + exercise 15 h preparation for exam 45 h self-study and homework (optionally in groups)	Verwendbarkeit M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Sprache English	Lehrende Andrea Herrmann	Prüfungsschema 1+1
Lernziele	The participants know advanced software engineering techniques which support decision-making during requirements prioritization, design, management decisions and risk management. The know how to manage knowledge in every day work life and got an introduction into decision theory.	
Lerninhalte	<p>knowledge management onthologies and Grounded Theory reverse engineering, code metrics learning organization storytelling decision-making and decision theory management decisions, business case risk management requirements prioritization decision-making in design: ATAM, SAAM, CBAM decision-making under uncertainty Mathematical Economics decision-making with several parties: Harvard concept, negotiations, Game Theory Decision Traps and Biases ethical decisions and machine ethics</p>	
Teilnahme- voraus- setzungen	recommended are: Einführung in Software Engineering (module ISW) or comparable competences	
Vergabe der LP und Modulendnote	The module is completed with a graded (oral or written) examination. The grade of the module is the grade of the examination. Prerequisite for the participation in the exam are 50% of the points for the homework.	
Nuetzliche Literatur	Raiffa, Howard; Richardson, John; Metcalfe, David: Negotiation analysis - the science and art of collaborative decision making, Belknap, Cambridge, 2002 or 2007	

4.4 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)
- Grundlagen der Optimierung (MD3)

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

- Grundmodul Numerik und Optimierung (MM15):

Nichtlineare Optimierung

Uncertainty Quantification 1

- Spezialisierungsmodul Numerik und Optimierung (MM35):

Computational Fluid Dynamics

Fundamentals of Computational Environmental Physics

Mathematical Methods of Image and Pattern Analysis II

Aus den Erganzungsmodulen:

- Berechenbarkeit und Komplexitat I
- Berechenbarkeit und Komplexitat II

4.5 Module aus dem M.Sc. Physik

Das Modul *Machine Learning and Physics* aus dem MSc Physik kann belegt werden. Für die Beschreibung wird auf das Modulhandbuch des MSc Physik verwiesen.

4.6 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.