

**Modulhandbuch**  
**Masterstudiengang**  
**„Angewandte Informatik“**

**Ruprecht-Karls-Universität Heidelberg**  
**Fakultät für Mathematik und Informatik**

**Fassung vom 09.02.2022 zur Prüfungsordnung vom 22.07.2010**  
**(mit letzter Änderung vom 07.02.2013)**

**Studienform:** Vollzeit

**Art des Studiengangs:** Konsekutiv

**Regelstudienzeit:** 4 Semester

**Anzahl der im Studiengang zu erwerbenden Leistungspunkte:** 120

**Studienstandort:** Heidelberg

**Anzahl der Studienplätze:** Keine Zulassungsbeschränkung

**Gebühren/Beiträge:** Gemäß allgemeiner Regelung der Universität Heidelberg

# Inhaltsverzeichnis

<b>1</b>	<b>Qualifikationsziele, Profil und Besonderheiten des Masterstudiengangs Angewandte Informatik</b>	<b>4</b>
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
<b>2</b>	<b>Studienverlaufsplan und Mobilität</b>	<b>8</b>
2.1	Studienverlaufsplan	8
2.2	Mobilitätsfenster	8
<b>3</b>	<b>Pflichtbereich</b>	<b>9</b>
	Wissenschaftliches Arbeiten	10
	Seminar	12
	Masterarbeit	13
	Anwendungsgebiet	14
<b>4</b>	<b>Wahlpflichtbereich</b>	<b>15</b>
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
	Computerspiele	35
	Discrete Structures 1	36
	Discrete Structures 2	37
	Fortgeschrittenenpraktikum	38
	Fundamentals of Machine Learning	39

Geometric Modeling and Animation	40
Hardware Aware Scientific Computing	42
Inverse Probleme	44
IT-Projektmanagement	45
IT-Sicherheit	47
Machine Learning	49
Mining Massive Datasets	51
Object-Oriented Programming for Scientific Computing	53
Optimization for Machine Learning	54
Praktische Geometrie	56
Projektseminar Biomedizinische Bildanalyse	57
Qualitätsmanagement	59
Randomisierte Algorithmen	60
Requirements Engineering	62
Scientific Visualization	63
Software Evolution	64
Software Ökonomie	66
Text Analytics	68
Visualisierung im Bereich Cultural Heritage	70
Volumenvisualisierung	71
Wissensmanagement und Entscheidungen im Software Engineering	72
4.4 Module aus dem B.Sc./M.Sc. Mathematik	73
4.5 Module aus dem M.Sc. Technische Informatik	74

# 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 forschungsnaher 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](http://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

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

### 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

<b>Code</b> IWA	<b>Name</b> Wissenschaftliches Arbeiten	
<b>LP</b> 2	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> jedes Wintersemester
<b>Format</b> 2 SWS	<b>Arbeitsaufwand</b> 60 h; davon 30 h Präsenzstudium 30 h Selbststudium und praktische Übungen (eventuell in Gruppen)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik
<b>Sprache</b> Deutsch oder Englisch	<b>Lehrende</b> wechselnd	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<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; kennen die Anforderungen an die Struktur von Anträgen zur Forschungsförderung.</p>	
<b>Lerninhalte</b>	<p>Literaturrecherche und -verwaltung Wissenschaftliches Vortragen, Schreiben, Publizieren und Begutachten Forschungsförderung über Drittmittel Forschungsmethoden und aktuelle Forschungsprojekte Wissenschaftliches Arbeiten nach dem Studium</p>	
<b>Teilnahmevoraussetzungen</b>	keine	
<b>Vergabe der LP und Modulendnote</b>	<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>	

<b>Nuetzliche Literatur</b>	
---------------------------------	--

## Seminar

<b>Code</b> IS	<b>Name</b> Seminar	
<b>LP</b> 4	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> jedes Semester
<b>Format</b> Seminar 2 + 2 SWS (Seminar/ Tutorium)	<b>Arbeitsaufwand</b> 120 h; davon 30 h Präsenzstudium 90 h Vorbereitung Vortrag	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik Lehramt Informatik M.Sc. Angewandte Informatik
<b>Sprache</b> Deutsch oder Englisch	<b>Lehrende</b> je nach Angebot	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<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>	
<b>Lerninhalte</b>	<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>	
<b>Teilnahmevoraussetzungen</b>	empfohlen sind: Kenntnisse im Themengebiet des Seminars	
<b>Vergabe der LP und Modulendnote</b>	<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>	
<b>Nuetzliche Literatur</b>		

## Masterarbeit

<b>Code</b> IMa	<b>Name</b> Masterarbeit	
<b>LP</b> 30	<b>Dauer</b> 6 Monate	<b>Angebotsturnus</b> jedes Semester
<b>Format</b> Betreutes Selbststudium 2 SWS, Kolloquium 1 SWS	<b>Arbeitsaufwand</b> 900 h; davon 810 h Bearbeitung eines individuellen Themas (Forschungs- und Entwicklungsarbeiten) und schriftliche Ausarbeitung 90 h Ausarbeitung Vortrag und Präsentation und Mitwirkung Kolloquium	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik
<b>Sprache</b> Deutsch oder Englisch	<b>Lehrende</b> je nach Angebot	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	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	
<b>Lerninhalte</b>	selbstständiges wissenschaftliches Bearbeiten einer beschränkten Aufgabenstellung aus der Informatik und ihren Anwendungen wissenschaftlicher Vortrag über die Problemstellung und die erarbeitete Lösung	
<b>Teilnahme- voraus- setzungen</b>	nach Prüfungsordnung mindestens 45 LP; weiterhin sind empfohlen: Wahlpflichtvorlesungen und Module Seminar (IS) und Fortgeschrittenenpraktikum (IFP)	
<b>Vergabe der LP und Mo- dulendnote</b>	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.	
<b>Nuetzliche Literatur</b>	wird von der Betreuerin bzw. dem Betreuer bekannt gegeben	

## Anwendungsgebiet

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

## 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)  
Computerspiele (ICS)  
Geometric Modeling and Animation (IGMA)  
Inverse Probleme (IIP)  
Praktische Geometrie (IPGeo)

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

### **Datenbanksysteme (DB)**

Complex Network Analysis (ICNA)  
Text Analytics (ITA)

### **Optimierung**

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

### **Parallele und Verteilte Systeme**

Mining Massive Datasets (IMMD)

### **Software Engineering (SWE)**

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

### **Technische Informatik**

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

### **Theoretische Informatik**

Algorithm Engineering (IAE)  
Algorithms and Data Structures 2 (IADS2)  
Berechenbarkeit und Komplexität I (MM41)  
Berechenbarkeit und Komplexität II (MM42)  
Discrete Structures 1 (IDS1)  
Discrete Structures 2 (IDS2)  
Randomisierte Algorithmen (IRA)

### **Wissenschaftliches Rechnen**

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

## 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

<b>Veranstaltung</b>	<b>Semester</b>	<b>LP</b>
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
<b>LP Summe</b>		<b>120</b>

## 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
<b>LP Summe</b>		120

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

## Vertiefung Information Systems Engineering

Diese Vertiefung befähigt zu Entwicklung, Betrieb und Wartung von komplexen Informationssystemen. Sie kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst die Seminare, Praktika, Bachelor- bzw. Masterarbeiten und die Wahlpflichtveranstaltungen. Dabei sind die Module zu gleichen Teilen auf die Lehr- und Forschungsgebiete der beiden Arbeitsgruppen Datenmanagement und -analyse (DMA) und Software Engineering (SWE) verteilt. Kernstück ist ein die beiden Lehrgebiete übergreifendes Praktikum (genannt ISE-Projekt) im Umfang von 16 LP, in dem im Team für externe Kunden ein Informationssystem mit ingenieurmäßigen Methoden unter Nutzung modernster Technologie entwickelt wird. Dazu gehören u.a. Konzepte und Methoden aus den Bereichen Daten- und Textanalyse, Informationsnetzwerke, Datenmanagement, Softwarequalität und Requirements Engineering. Das ISE-Projekt wird immer im Wintersemester angeboten. Ansprechpartner für diese Vertiefung sind Herr Professor Dr. Gertz (DMA) und Frau Professor Dr. Paech (SWE). Vor Beginn der Vertiefung wird ein Beratungsgespräch bei einem der Ansprechpartner empfohlen.

Vertiefende Module aus dem Gebiet SWE:

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

Vertiefende Module aus dem Gebiet DMA:

- Complex Network Analysis (ICNA) 8 LP
- Text Analytics (ITA) 8 LP

Ergänzende Module:

- Artificial Intelligence for Programming (IAIP) 6 LP
- Fundamentals of Machine Learning (IFML) 8 LP
- IT-Sicherheit (IITS) 8 LP
- Machine Learning (IML) 8 LP
- Mining Massive Datasets (IMMD) 6 LP
- Object-Oriented Programming for Scientific Computing (IOPSC) 6 LP
- Scientific Visualization (ISV) 8 LP

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
Seminar - Lehrgebiet SWE	1 - 3	4
Seminar - Lehrgebiet DMA	1 - 3	4
ISE-Projekt	1 - 3	16
2 Vertiefende Module Lehrgebiet SWE	1 - 3	16
2 Vertiefende Module Lehrgebiet DMA	1 - 3	16
Weitere Module aus SWE oder DMA	1 - 3	6
Ergänzende Module 1	1 - 3	8
Masterarbeit SWE oder DMA	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		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
<b>LP Summe</b>		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
<b>LP Summe</b>		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

<b>Code</b> IAML	<b>Name</b> Advanced Machine Learning	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> follows *Fundamentals of Machine Learning*
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240h, thereof 60h lecture 90h tutorials, homework, lecture wrap-up 90h graded final report	<b>Verwendbarkeit</b> cannot be combined with *Machine Learning* M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Ullrich Köthe	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	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.	
<b>Lerninhalte</b>	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)	
<b>Teilnahmevoraussetzungen</b>	recommended are: lecture *Fundamentals of Machine Learning* or similar	
<b>Vergabe der LP und Modulendnote</b>	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.	
<b>Nuetzliche Literatur</b>	David Barber: Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012 Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, 2006	

## Algorithm Engineering

<b>Code</b> IAE	<b>Name</b> Algorithm Engineering	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every summer semester
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240h; thereof 90h lectures and tutorials, 15h exam preparations, 135h lecture wrap-up and homework	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Christian Schulz	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<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>	
<b>Lerninhalte</b>	<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>	
<b>Teilnahmevoraussetzungen</b>	<p>recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Mathematik für Informatiker 1 oder Lineare Algebra 1 (MA4), Algorithms and Data Structures 2</p>	
<b>Vergabe der LP und Modulendnote</b>	<p>The module is completed with a graded oral exam. 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.</p>	

<b>Nuetzliche Literatur</b>	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

<b>Code</b> IADS2	<b>Name</b> Algorithms and Data Structures 2	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every winter semester
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240h; thereof 90h lectures and tutorials, 15h exam preparations, 135h lecture wrap-up and homework	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Christian Schulz	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>Students:</p> <ul style="list-style-type: none"> <li>- understand fundamental theoretical and practical concepts of advanced algorithms and data structures,</li> <li>- get to know established methods and algorithms,</li> <li>- are familiar with issues of efficient implementations,</li> <li>- are able to identify/formulate algorithmic problems in/for different application areas,</li> <li>- are able to analyse new algorithms as well as analysing their running time, and select appropriate algorithms for applications</li> <li>- are able to apply algorithms and data structures to real-world problems, and can objectively assess the quality of the results</li> </ul>	
<b>Lerninhalte</b>	<p>Introduction to Algorithm Engineering:</p> <ul style="list-style-type: none"> <li>- advanced data structures (efficient addressable priority queues, monotone priority queues, external priority queues),</li> <li>- advances graph algorithms (strongly connected components, shortest paths, maximum flows / min s-t cuts, min-cost flows),</li> <li>techniques to solve problems to optimality (branch-and-bound, branch-and-reduce, dynamic programming, integer linear programming as a modelling tool),</li> <li>- introduction to randomized algorithms, greedy algorithms, approximation algorithms, advanced string algorithms, geometric algorithms, external memory algorithms</li> </ul>	
<b>Teilnahmevoraussetzungen</b>	<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>	
<b>Vergabe der LP und Modulendnote</b>	<p>The module is completed with a graded oral exam. 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.</p>	

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

## Artificial Intelligence for Programming

<b>Code</b> IAIP	<b>Name</b> Artificial Intelligence for Programming	
<b>LP</b> 6	<b>Dauer</b> one semester	<b>Angebotsturnus</b> at least every 4th semester
<b>Format</b> Lecture 2 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 180 h; thereof 60 h lecture 15 h preparation for exam 105 h self-study and working on assignments (optionally in groups)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Artur Andrzejak	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>Expected learning outcomes are:</p> <ul style="list-style-type: none"> <li>- Knowledge of selected classical methods in artificial intelligence, in particular knowledge representation, search methods, rule systems</li> <li>- Basic knowledge about probabilistic models and probabilistic programming</li> <li>- Knowledge of techniques for code representation and parsing</li> <li>- Knowledge of techniques for modeling of code via neural networks</li> <li>- Knowledge of basic and advanced methods for program synthesis</li> <li>- Familiarity with semantic parsing and code summarization</li> <li>- Familiarity with selected applications of AI for programming, e.g. code-to-code translation, code recommendations, and detection of bugs in code.</li> </ul>	
<b>Lerninhalte</b>	<p>This module covers the following topics:</p> <ul style="list-style-type: none"> <li>- introduction to classical methods in artificial intelligence, in particular knowledge representation, search methods, rule systems</li> <li>- introduction to probabilistic models and probabilistic programming</li> <li>- fundamentals of code representation and parsing</li> <li>- modeling of code via neural networks and sequence models/transformers</li> <li>- basic and advanced methods for program synthesis</li> <li>- introduction to semantic parsing and code summarization</li> <li>- state-of-the-art applications of AI for programming, e.g. code-to-code translation, code recommendations, detection of vulnerabilities in code.</li> </ul>	
<b>Teilnahmevoraussetzungen</b>	<p>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.</p>	
<b>Vergabe der LP und Modulendnote</b>	<p>The module is completed with a graded oral or written exam. 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.</p>	

<b>Nuetzliche Literatur</b>	<p>Stuart J. Russell: Artificial intelligence: a modern ap-proach, (3rd ed.), Pearson, 2016, Heidi: <a href="https://bit.ly/2V9LQT9">https://bit.ly/2V9LQT9</a></p> <p>Noah D. Goodman, Joshua B. Tenenbaum: Probabil-istic Models of Cognition (2nd ed.), 2016. Online: <a href="https://probmods.org/">https://probmods.org/</a></p> <p>Jeremy Howard: Deep learning for coders with fastai and PyTorch, (1st ed.), O'Reilly, 2020, Online via Heidi: <a href="https://bit.ly/3jUMkH7">https://bit.ly/3jUMkH7</a></p> <p>Aurélien Géron: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, (2nd ed.), O'Reilly, 2019, Online via Heidi: <a href="https://bit.ly/3dVhieA">https://bit.ly/3dVhieA</a></p>
---------------------------------	---

## Complex Network Analysis

<b>Code</b> ICNA	<b>Name</b> Complex Network Analysis	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every 2nd wintersemester
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h lecture 12 h preparation for exam 130 h self-study and working on assignments/projects (optionally in groups)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing B.Sc. Mathematik
<b>Sprache</b> English	<b>Lehrende</b> Michael Gertz	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>Students</p> <ul style="list-style-type: none"> <li>- can describe basic measures and characteristics of complex networks</li> <li>- can implement and apply basic network analysis algorithms using programming environments such as R or Python</li> <li>- can describe different network models and can describe, compute, and analyze characteristic parameters of these models</li> <li>- know how to compute different complex network measures and how to interpret these measures</li> <li>- know different generative models for constructing complex networks, especially scale-free networks</li> <li>- know the fundamental methods for the detection of communities in networks and the analysis of their evolution over time</li> <li>- are familiar with basic concepts of network robustness</li> <li>- understand the principles behind the spread of phenomena in complex networks</li> </ul>	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Graph theory and graph algorithms; basic network measures</li> <li>- Random networks and their characteristics (degree distribution, component sizes, clustering coefficient, network evolution), small world phenomena</li> <li>- Scale-free property of networks, power-laws, hubs, universality</li> <li>- Barabasi-Albert model, growth and preferential attachment, degree dynamics, diameter and clustering coefficient</li> <li>- Evolving networks, Bianconi-Barabasi model, fitness, Bose-Einstein condensation</li> <li>- Degree correlation, assortativity, degree correlations, structural cutoffs</li> <li>- Network robustness, percolation theory, attack tolerance, cascading failures</li> <li>- Communities, modularity, community detection and evolution</li> <li>- Spreading phenomena, epidemic modeling, contact networks, immunization, epidemic prediction</li> </ul>	
<b>Teilnahmevoraussetzungen</b>	recommended are: Algorithmen und Datenstrukturen (IAD), Knowledge Discovery in Databases (IKDD), Lineare Algebra I (MA4)	

<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded written exam. 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.
<b>Nuetzliche Literatur</b>	<ul style="list-style-type: none"> <li>- Albert-Laszlo Barabasi: Network Science, Cambridge University Press, 2016.</li> <li>- M.E.J. Newmann: Networks: An Introduction, Oxford University Press, 2010.</li> <li>- Vito Latora, Vincenzo Nicosia, Giovanni Russo: Complex Networks - Principles, Methods and Applications, Cambridge University Press, 2017.</li> <li>- David Easley, Jon Kleinberg: Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.</li> <li>- Stanley Wasserman, Katherine Faust: Social Network Analysis-Methods and Applications, Cambridge University Press, 1994.</li> </ul>

## Computational Geometry

<b>Code</b> ICGeo	<b>Name</b> Computational Geometry	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> irregular
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 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)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Susanne Krömker	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<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>	
<b>Lerninhalte</b>	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	
<b>Teilnahmevoraussetzungen</b>	recommended is: Algorithmen und Datenstrukturen (IAD)	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded oral exam. 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.	
<b>Nützliche Literatur</b>	<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

<b>Code</b> ICG	<b>Name</b> Computer Graphics	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every 3rd semester
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h on-campus program 15 h exam preparation 135 h independent study and exercises (possibly in groups)	<b>Verwendbarkeit</b> 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
<b>Sprache</b> English	<b>Lehrende</b> Filip Sadlo	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	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.	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Introduction</li> <li>- Perception and Color</li> <li>- Raytracing</li> <li>- Transformations</li> <li>- Rasterization</li> <li>- OpenGL</li> <li>- Textures</li> <li>- Spatial Data Structures</li> </ul>	
<b>Teilnahmevoraussetzungen</b>	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded written or oral exam. 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.	
<b>Nuetzliche Literatur</b>	<p>P. Shirley, S. Marschner: Fundamentals of Computer Graphics, 3rd Edition, AK Peters</p> <p>OpenGL Specifications(GL 4.5 + GLSL 4.50) <a href="http://www.opengl.org/registry/">http://www.opengl.org/registry/</a> 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>	

## Computerspiele

<b>Code</b> ICS	<b>Name</b> Computerspiele	
<b>LP</b> 8	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> jedes Sommersemester
<b>Format</b> Vorlesung 3 SWS + Übung 3 SWS	<b>Arbeitsaufwand</b> 240 h; davon 75 h Präsenzstudium 15 h Prüfungsvorbereitung 150 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> Englisch	<b>Lehrende</b> Jürgen Hesser	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	Die Studierenden lernen die Konzepte von der informatischen Sicht kennen, was Computerspiele ausmacht und welche Herausforderungen damit verbunden sind. Sie lernen, wie man gute und effiziente Architekturkonzepte dafür entwickelt sowie wie man typische Probleme aus Graphik, Kollisionserkennung, Animation/Physik, Pfadplanung/KI umsetzt. Zudem lernen sie in den Übungen, wie man konkrete Spiele entwickelt, so dass sie in der Lage sind, eigene Spieleengines zu realisieren.	
<b>Lerninhalte</b>	Überblick über die Einteilung von Computerspielen Architektur von Game Engines Vorstellung von OGRE als einer open-source Game Engine Graphik und Computerspiele: ein Überblick Kollisionserkennungstechniken Animationstechniken und Physik bei Computerspielen mit Fokus auf der open source Bibliothek Bullet Pfadplanung und KI	
<b>Teilnahme- voraus- setzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Vergabe der LP und Mo- dulendnote</b>	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.	
<b>Nuetzliche Literatur</b>	Gregory et al: Game Engine Architecture Ericson: Real-Time Collision Detection Eberly: Game Physics Millington: Artificial Intelligence for Games	

## Discrete Structures 1

<b>Code</b> IDS1	<b>Name</b> Discrete Structures 1	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every winter semester
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h lecture 20 h preparation for exam 130 h self-study and working on assignments/projects (optionally in groups)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik B.Sc. Informatik B.Sc. Mathematik
<b>Sprache</b> English	<b>Lehrende</b> Felix Joos	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>Students</p> <ul style="list-style-type: none"> <li>- understand several basic graph parameters and the central theorems in these areas</li> <li>- can solve easy problems involving discussed topics</li> <li>- can describe graph algorithms computing discussed graph parameters</li> <li>- know how to use graphs and graph parameters to model real world problems</li> </ul>	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Introduction to graph theory terminology</li> <li>- Matchings in graph and hypergraphs</li> <li>- Graph connectivity</li> <li>- Planar graphs</li> <li>- Graph Colouring</li> <li>- Hamilton Cycles</li> <li>- Ramsey Theory</li> <li>- Random graphs</li> <li>- Algebraic Graph constructions (Cayley graphs, Kneser graphs,...)</li> <li>- Algorithms computing discussed graph parameters</li> </ul>	
<b>Teilnahmevoraussetzungen</b>	recommended are: Einführung in die Praktische Informatik (IPI), Mathematik für Informatiker 1 (IMI1) or Lineare Algebra 1 (MA4), Mathematik für Informatiker 2 (IMI2) oder Analysis 1 (MA1)	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded oral or written exam. 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.	
<b>Nuetzliche Literatur</b>	<ul style="list-style-type: none"> <li>- Reinhard Diestel Graph Theory, 5th edition, Springer, 2016/17</li> <li>- Douglas West, Introduction to Graph Theory, Pearson, 2011.</li> <li>- J.A. Bondy and U.S.R. Murty, Graph Theory, Springer, 2008.</li> <li>- Bernhard Korte and Jens Vygen, Combinatorial Optimization, 6th edition, 2018.</li> </ul>	

## Discrete Structures 2

<b>Code</b> IDS2	<b>Name</b> Discrete Structures 2	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> irregularly in the summer semester
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h lecture 20 h preparation for exam 130 h self-study and working on assignments/projects (optionally in groups)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Mathematik
<b>Sprache</b> English	<b>Lehrende</b> Felix Joos	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>Students</p> <ul style="list-style-type: none"> <li>- understand several advanced graph parameters and the central theorems in these areas</li> <li>- can solve problems involving discussed topics</li> <li>- can reprove the central considered results</li> </ul>	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Probabilistic Methods</li> <li>- Extremal graph theory</li> <li>- Expander graphs</li> <li>- Quasirandom graphs</li> <li>- Further advanced topics</li> </ul>	
<b>Teilnahmevoraussetzungen</b>	recommended is: Discrete Structures 1	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded oral or written exam. 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.	
<b>Nuetzliche Literatur</b>	<ul style="list-style-type: none"> <li>- Reinhard Diestel Graph Theory, 5th edition, Springer, 2016/17</li> <li>- Douglas West, Introduction to Graph Theory, Pearson, 2011.</li> <li>- J.A. Bondy and U.S.R. Murty, Graph Theory, Springer, 2008.</li> <li>- Bernhard Korte and Jens Vygen, Combinatorial Optimization, 6th edition, 2018.</li> </ul>	

## Fortgeschrittenenpraktikum

<b>Code</b> IFP	<b>Name</b> Fortgeschrittenenpraktikum	
<b>LP</b> 8	<b>Dauer</b>	<b>Angebotsturnus</b> jedes Semester
<b>Format</b> Praktikum 6 SWS	<b>Arbeitsaufwand</b> 240 h; davon mind. 25 h Präsenzzeit 10 h Vorbereitung Vortrag	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik B.Sc. Informatik Lehramt Informatik
<b>Sprache</b> Deutsch oder Englisch	<b>Lehrende</b> je nach Angebot	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<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>	
<b>Lerninhalte</b>	<p>Domänenkenntnisse abhängig von den Lehrenden; allgemeine Lerninhalte sind: Vertiefung in die Projektarbeit Eigenständige Entwicklung von komplexer Software und deren Dokumentation</p>	
<b>Teilnahmevoraussetzungen</b>	empfohlen sind: Anfängerpraktikum (IAP), Einführung in Software Engineering (ISW)	
<b>Vergabe der LP und Modulendnote</b>	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.	
<b>Nützliche Literatur</b>		

## Fundamentals of Machine Learning

<b>Code</b> IFML	<b>Name</b> Fundamentals of Machine Learning	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> in (irregular) alternation with *Machine Learning*
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240h, thereof 60h lecture 90h tutorials, homework, lecture wrap-up 90h graded final report	<b>Verwendbarkeit</b> cannot be combined with *Machine Learning* M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Ullrich Köthe	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	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.	
<b>Lerninhalte</b>	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)	
<b>Teilnahmevoraussetzungen</b>	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
<b>Vergabe der LP und Modulendnote</b>	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.	
<b>Nuetzliche Literatur</b>	Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning (2nd edition), Springer, 2009	

## Geometric Modeling and Animation

<b>Code</b> IGMA	<b>Name</b> Geometric Modeling and Animation	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every 3rd semester
<b>Format</b> Lecture 4 SWS + Exercise 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h on-campus program 15 h exam preparation 135 h independent study and exercises (possibly in groups)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Filip Sadlo	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<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>	
<b>Lerninhalte</b>	<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>	
<b>Teilnahme- voraus- setzungen</b>	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Vergabe der LP und Mo- dulendnote</b>	The module is completed with a graded oral or written exam. 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.	

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

## Hardware Aware Scientific Computing

<b>Code</b> IHASC	<b>Name</b> Hardware Aware Scientific Computing	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> irregular
<b>Format</b> Lecture 4 SWS + Exercise Course 2 SWS	<b>Arbeitsaufwand</b> 240h;thereof 90h lecture 15h preparation for exam 135h self-study and working on assignments/projects (optionally in groups)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Peter Bastian	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	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.	
<b>Lerninhalte</b>	<p>Parallel Computer Architecture</p> <ul style="list-style-type: none"> <li>- Pipelining and super-scalar processors, SIMD vectorisation</li> <li>- Caches</li> <li>- Multicore architectures</li> <li>- GPUs</li> <li>- Communication networks</li> </ul> <p>Programming Models</p> <ul style="list-style-type: none"> <li>- Shared memory programming with OpenMP and C++ threads</li> <li>- OpenCL or Cuda</li> <li>- Task-based programming</li> <li>- Message-passing, MPI</li> </ul> <p>Parallel Algorithms</p> <ul style="list-style-type: none"> <li>- Speedup &amp; scalability</li> <li>- Roofline model</li> <li>- Linear Algebra: Matrix-Vector, Matrix multiplication, solving dense systems, solving sparse systems</li> <li>- Iterative Solution of Linear Systems</li> <li>- High-Performance Libraries</li> <li>- Differential equations</li> <li>- Particle Methods</li> </ul>	
<b>Teilnahmevoraussetzungen</b>	basic knowledge in computer architecture and numerical methods; good programming skills in C++	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded exam. The note of this exam gives the note for this module. Details for this exam as well as the requirements for the assignment of credits will be given by the lecturer at the beginning of this course.	

<b>Nuetzliche Literatur</b>	Frédéric Magoules, François-Xavier Roux, Guillaume Houzeaux: Parallel Scientific Computing, Wiley, 2016, doi: 10.1002/9781118761687
---------------------------------	---

## Inverse Probleme

<b>Code</b> IIP	<b>Name</b> Inverse Probleme	
<b>LP</b> 8	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> jedes Sommersemester
<b>Format</b> Vorlesung 2 SWS, Übung 2 SWS und Hausarbeiten	<b>Arbeitsaufwand</b> 240 h; davon 60 h Präsenzstudium 15 h Prüfungsvorbereitung 165 h Selbststudium und Aufgabenbearbeitung, Hausarbeiten	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
<b>Sprache</b> Englisch	<b>Lehrende</b> Jürgen Hesser	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	Die Studierenden verstehen, was inverse Probleme sind und warum sie schwer zu lösen sind. Sie lernen die Prinzipien kennen, wie man sowohl deterministische Probleme, als auch stochastische Probleme löst und dabei auch die Regularisierungsparameter geeignet wählt. Schließlich erfahren sie die neuesten Entwicklungen im Bereich compressed sensing. Alle Prinzipien werden an zwei ausgewählten Gebieten, der Tomographie und des Deblurrings dargestellt. Sie erhalten damit die Kompetenz komplexe Probleme zu lösen, die mit klassischen Techniken nicht stabil lösbar sind und sind damit in der Lage auch komplexe experimentelle Messungen adäquat auswerten zu können.	
<b>Lerninhalte</b>	Deterministische inverse Probleme Stochastische inverse Probleme Wahl der Regularisierungsparameter Compressed sensing Tomographie Deblurring	
<b>Teilnahme- voraus- setzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Numerische Mathematik	
<b>Vergabe der LP und Mo- dulendnote</b>	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.	
<b>Nuetzliche Literatur</b>	M. Bertero, P. Boccacci: Introduction to Inverse Problems in Imaging, IoP, 2002  web-Page and book: <a href="http://www.slaney.org/pct/pct-toc.html">http://www.slaney.org/pct/pct-toc.html</a>	

## IT-Projektmanagement

<b>Code</b> IPM	<b>Name</b> IT-Projektmanagement	
<b>LP</b> 3	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> jedes 2. Wintersemester
<b>Format</b> Vorlesung+Übung 2 SWS	<b>Arbeitsaufwand</b> Vorlesung + Übung 90 h insgesamt, davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)  für eine Anrechnung als EPG zusätzlich: 90 h Forschungsprojekt, davon 15 h Einarbeitung und Literaturrecherche 15 h Vorbereitung einer Studie, z.B. Interview-Training 20 h Durchführung einer Studie 20 h Auswertung 20 h Abschlusspräsentation + Berichterstellung	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
<b>Sprache</b> Englisch	<b>Lehrende</b> Andrea Herrmann	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	Die Teilnehmer/innen können ein Projekt planen und überwachen, verstehen, wie Projekte in Organisationen eingebettet sind und haben Grundkenntnisse in vertraglichen Themen. Die Teilnehmer/innen des EPG können ein Forschungsprojekt selbständig durchführen und kennen Forschungsergebnisse aus der Gender-Forschung.	
<b>Lerninhalte</b>	Projektplanung, Projektorganisation Kostenschätzung Angebot/ Vertrag, Verhandeln Vorgehensmodelle Risikomanagement Controlling IT-Vertragsrecht Änderungsmanagement Zeitmanagement Projektabschluss Verteilte Softwareentwicklung	
<b>Teilnahmevoraussetzungen</b>	keine	

<b>Vergabe der LP und Modulendnote</b>	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.
<b>Nuetzliche Literatur</b>	PMI (Project Management Institute): A Guide to the Project Management Body of Knowledge (PM BOK ® Guide), 4. Ausgabe 2008

## IT-Sicherheit

<b>Code</b> IITS	<b>Name</b> IT-Sicherheit	
<b>LP</b> 8	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> unregelmäßig
<b>Format</b> Vorlesung 4 SWS + Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
<b>Sprache</b> Deutsch oder Englisch	<b>Lehrende</b> Vincent Heuveline	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>Studierende</p> <ul style="list-style-type: none"> <li>* erwerben umfangreiches Wissen über die Funktionsweise und Verwundbarkeiten vernetzter Computersysteme, und können somit Konzepte zur IT-Netzicherheit bewerten und entwerfen</li> <li>* erlangen grundlegende Kenntnisse über die Sicherung großer Netzwerke und der Kommunikationsinfrastruktur (Routing, Namensauflösung, Internet-Firewalls, Intrusion Detection Systeme)</li> <li>* erwerben Kenntnisse im Bereich Kryptographie: Theorie der Kryptographie und praktische Umsetzung typischer kryptographischer Verfahren im Zusammenhang mit kryptographischen Prüfverfahren, symmetrischen und asymmetrischen Chiffrierverfahren</li> <li>* erwerben umfassende Kompetenzen zur Detektion von Cyberangriffen; grundlegende Kompetenzen im Feld der IT-/Cyber-Forensik</li> <li>* erwerben praktische Erfahrungen bei der Verwendung von dedizierter Software zur Detektion von Angriffsszenarien im Datennetz</li> <li>* erwerben grundlegender Kenntnisse zum BSI IT-Grundschutzkonzept</li> <li>* erwerben grundlegender Kenntnisse im Penetrationstesting und dazugehöriger Werkzeuge</li> </ul> <p>Langfristiges Ausbildungsziel: Einsatz-/Beschäftigungsfähigkeit in der Breite des Arbeitsfeldes IT-Sicherheit;</p>	

<b>Lerninhalte</b>	<p>Der IT-Sicherheit kommt bei der allgegenwärtigen Digitalisierung eine Schlüsselrolle zu. Diese Vorlesung vermittelt methodische Ansätze zur Modellierung und Bewertung von Angriffsszenarien, auf Basis welcher wirksame technische Gegenmaßnahmen umgesetzt werden können. Insbesondere werden folgende Schwerpunkte adressiert:</p> <ul style="list-style-type: none"> <li>- Sicherheitsmodelle und Bewertungskriterien</li> <li>- Kryptographische Prüfwerte: Modifikationserkennungs- und Nachrichtenauthentisierungswerte</li> <li>- Symmetrische und asymmetrische kryptographische Verfahren</li> <li>- Kryptographische Protokolle</li> <li>- Zugangs/Zutritts/Zugriffskontrolle</li> <li>- Schutz von Kommunikationsinfrastrukturen</li> <li>- Digitale Identität</li> <li>- Penetrationstesting / Schwachstellenanalyse</li> <li>- BSI IT-Grundschutzbausteine</li> </ul> <p>Mit Hilfe von virtuellen Maschinen in einem geschützten Bereich werden klassische Angriffs- und Schutzszenarien praktisch untersucht. Hier werden sog. Experimentierblätter samt Daten zur Verfügung gestellt.</p>
<b>Teilnahmevoraussetzungen</b>	<p>empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Betriebssysteme und Netzwerke (IBN), Einführung in die Technische Informatik (ITE), Algorithmen und Datenstrukturen (IAD)</p>
<b>Vergabe der LP und Modulendnote</b>	<p>Das Modul wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.</p>
<b>Nuetzliche Literatur</b>	<p>C. Eckert, IT-Sicherheit: Konzepte, Verfahren, Protokolle De Gruyter Studium. Oldenbourg: de Gruyter, 2014.  T.W. Harich, IT-Sicherheitsmanagement: Arbeitsplatz IT Security Manager. MITP, 2012.  J.P.Müller, Security for Web Developers, O'Reilly, 2018</p>

## Machine Learning

<b>Code</b> IML	<b>Name</b> Machine Learning	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> in (irregular) alternation with *Fundamentals of Machine Learning* + *Advanced Machine Learning*
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> Arbeitsaufwand: 240h, thereof 60h lecture 90h tutorials, homework, lecture wrap-up 90h graded final report	<b>Verwendbarkeit</b> 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
<b>Sprache</b> English	<b>Lehrende</b> Ullrich Köthe	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	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.	
<b>Lerninhalte</b>	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	
<b>Teilnahmevoraussetzungen</b>	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
<b>Vergabe der LP und Modulendnote</b>	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.	

<b>Nuetzliche Literatur</b>	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

<b>Code</b> IMMD	<b>Name</b> Mining Massive Datasets	
<b>LP</b> 6	<b>Dauer</b> one semester	<b>Angebotsturnus</b> at least every 4th semester
<b>Format</b> Lecture 2 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 180 h; thereof 60 h lecture 15 h preparation for exam 105 h self-study and working on assignments (optionally in groups)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Artur Andrzejak	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<ul style="list-style-type: none"> <li>* Knowledge of selected approaches and programming paradigms of parallel data processing</li> <li>* Knowledge how to use tools for parallel data processing (among others Apache Hadoop and Spark)</li> <li>* Familiarity with application domains of big data analysis</li> <li>* Knowledge of methods of parallel pre-processing of data</li> <li>* Knowledge of methods like classification, regression, clustering and their parallel implementations</li> <li>* Knowledge of scaling of parallel algorithms</li> </ul>	
<b>Lerninhalte</b>	<p>This module covers the following topics:</p> <ul style="list-style-type: none"> <li>* programming paradigms for parallel-distributed data processing, especially Map-Reduce and Spark programming models</li> <li>* usage of tools like Apache Spark, Hadoop, Pig, Hive, and possibly other frameworks for parallel-distributed data processing</li> <li>* application cases in parallel data analysis, for example clustering, recommendation, search for similar objects, mining of data streams</li> <li>* techniques for parallel pre-processing of data</li> <li>* fundamentals of analysis techniques such as classification, regression, clustering and evaluation of the results</li> <li>* parallel algorithms for data analysis and their implementations</li> <li>* theory and practice of scalability and tuning of frameworks</li> </ul>	
<b>Teilnahmevoraussetzungen</b>	recommended are Knowledge of Java/Python and in elementary probability theory / statistics; module IBD can be taken as a complement / extension.	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded exam. This note of this exam gives the note for this module. Details for this exam as well as the requirements for the assignment of credits will be given by the lecturer at the beginning of this course.	

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

## Object-Oriented Programming for Scientific Computing

<b>Code</b> IOPSC	<b>Name</b> Object-Oriented Programming for Scientific Computing	
<b>LP</b> 6	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every summer semester
<b>Format</b> Lecture 2 SWS + Exercise on computer 2 SWS	<b>Arbeitsaufwand</b> 180 h; thereof 60 h lecture 105 h self-study and working on assignments 15 h preparation for exam	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Ole Klein	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	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.	
<b>Lerninhalte</b>	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	
<b>Teilnahmevoraussetzungen</b>	recommended are: Einführung in die Praktische Informatik (IPI), alternatively basic knowledge of an object-oriented programming language	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded written exam. The final grade of the module is determined by the grade of the exam. The lecture will give the requirements for the assignment of credits.	
<b>Nuetzliche Literatur</b>		

## Optimization for Machine Learning

<b>Code</b> IOML	<b>Name</b> Optimization for Machine Learning	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every winter semester
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 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)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Mathematik M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Bogdan Savchynskyy	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>The students</p> <ul style="list-style-type: none"> <li>- can analyze optimization methods for machine learning problems and estimate the area of their potential application</li> <li>- can competently apply existing algorithms and program packages for inference and learning with graphical models and neural networks</li> <li>- know typical optimization techniques for inference and learning with graphical models and neural networks</li> <li>- understand the basics of convex analysis, convex optimization, convex duality theory, (integer) linear programs and their geometry</li> </ul>	
<b>Lerninhalte</b>	<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"> <li>- 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</li> <li>- Graphical Models: dynamic programming, sub-gradient and block-coordinate ascent inference methods, min-cut/max-flow based inference, structured risk minimization for graphical models</li> <li>- neural networks: architectures, backpropagation algorithm, stochastic gradient descent and its variants for training neural networks.</li> </ul>	
<b>Teilnahmevoraussetzungen</b>	recommended are: linear algebra, analysis and any universal programming language (e.g. C/C++/Pascal/python)	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded oral exam. 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.	

<b>Nuetzliche Literatur</b>	will beannounced by the lecturer at the beginning of the course
---------------------------------	---

## Praktische Geometrie

<b>Code</b> IPGeo	<b>Name</b> Praktische Geometrie	
<b>LP</b> 4	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> unregelmäßig
<b>Format</b> Vorlesung 2 SWS, Übung 1 SWS	<b>Arbeitsaufwand</b> 120 h; davon 45 h Präsenzstudium 60 h Aufgabenbearbeitung 15 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> Deutsch	<b>Lehrende</b> Susanne Krömker	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	Verständnis grundlegender geometrischer Konzepte zur Datenanalyse sowie effektive Punktssuche und Weiterverarbeitung von Messdaten Souveräner Umgang mit Projektionen und Beschreibungen jenseits der dreidimensionalen Erfahrungswelt Berechnung geometrischer Invarianten, Distanzen, Krümmungen aus Messdaten, rekonstruierten und generierten Flächen	
<b>Lerninhalte</b>	Grundlegende Gebiete der Geometrie mit Relevanz in Computergraphik, Bildverarbeitung, Mustererkennung, Computer Vision und Geometrischem Modellieren (i) Analytische Geometrie: Operationen auf Vektorräumen mit geeigneten Koordinaten und Abbildungen (Affinitäten, Kollinearitäten), geometrische Ausgleichsprobleme aus fehlerbehafteten Messdaten (ii) Projektive Geometrie: Zentralprojektion und inverse Rekonstruktion von 3D-Objekten aus ebenen Bildern (Computer Vision, Geodäsie), Unterschiede zwischen B-Spline-Kurven und -Flächen und der Klasse der NURBS, Freiformgeometrien in CAD-Systemen (iii) Differentialgeometrie: Parameterdarstellungen in der geometrischen Datenverarbeitung, implizite Darstellungen (level sets), Abschätzung von Invarianten aus diskreten Daten (Triangulierungen, Punktwolken)	
<b>Teilnahmevoraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Mathematik für Informatiker (IMI1 und 2) oder Lineare Algebra (MA4)	
<b>Vergabe der LP und Modulendnote</b>	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.	
<b>Nützliche Literatur</b>	Geometrie für Informatiker, Skript TU Wien 2004, Helmut Pottmann Aktuelle Fachveröffentlichungen	

## Projektseminar Biomedizinische Bildanalyse

<b>Code</b> IPBB	<b>Name</b> Projektseminar Biomedizinische Bildanalyse	
<b>LP</b> 6	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> jedes Sommersemester
<b>Format</b> 2 Teile Seminar und Projekt, 4 SWS	<b>Arbeitsaufwand</b> 180 h (je zur Hälfte Seminar und Projekt) 60 h Präsenzstudium 120 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
<b>Sprache</b> Deutsch	<b>Lehrende</b> Karl Rohr	<b>Prüfungsschema</b>
<b>Lernziele</b>	<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>	
<b>Lerninhalte</b>	<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 Seminaranteil (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>	
<b>Teilnahmevoraussetzungen</b>	empfohlen sind: Grundkenntnisse in Bildverarbeitung (Computer Vision, Image Analysis), Programmierkenntnisse, Kenntnisse in Software Engineering	
<b>Vergabe der LP und Modulendnote</b>	<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>	

<b>Nuetzliche Literatur</b>	Bekanntgabe in der Lehrveranstaltung
---------------------------------	--------------------------------------

## Qualitätsmanagement

<b>Code</b> ISWQM	<b>Name</b> Qualitätsmanagement	
<b>LP</b> 8	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> jedes 2. Sommersemester
<b>Format</b> Vorlesung 3 SWS + Übung 3 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung im Team	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Angewandte Informatik
<b>Sprache</b> Deutsch	<b>Lehrende</b> Barbara Paech	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	Dieses Modul vertieft die Grundkenntnisse und -fähigkeiten des Software Engineering insbesondere im Bereich des Qualitätsmanagement. Kenntnis der unter Inhalt angegebenen Methoden, Prozess und Werkzeuge Fähigkeit, Qualitätssicherung, Qualitätsmanagement und Prozessverbesserung und -management einem Softwareentwicklungsprojekt durchzuführen bzw unter Anleitung angewandte Forschung dazu durchzuführen. Fähigkeit Teilaufgaben im Team durchzuführen (eventuell mit *echten* Kunden)	
<b>Lerninhalte</b>	Methoden, Prozesse und Werkzeuge für Qualitätssicherung Qualitätsmanagement Verbesserung von Softwareentwicklungsprozessen Management von Softwareentwicklungsprozessen	
<b>Teilnahme- voraus- setzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden	
<b>Vergabe der LP und Mo- dulendnote</b>	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.	
<b>Nuetzliche Literatur</b>	Wird jährlich aktualisiert	

## Randomisierte Algorithmen

<b>Code</b> IRA	<b>Name</b> Randomisierte Algorithmen	
<b>LP</b> 6	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> mindst. jedes 4. Semester
<b>Format</b> Vorlesung 3 SW + Übung 1 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 40 h Prüfungsvorbereitung 80 h Selbststudium und Bearbeitung der Übungsaufgaben (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. Scientific Computing
<b>Sprache</b> Deutsch	<b>Lehrende</b> Wolfgang Merkle	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	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.	
<b>Lerninhalte</b>	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	
<b>Teilnahmevoraussetzungen</b>	empfohlen sind: elementare Grundkenntnisse in Algorithmen wie sie z.B. im Modul Algorithmen und Datenstrukturen (IAD) vermittelt werden.	
<b>Vergabe der LP und Modulendnote</b>	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.	

<b>Nuetzliche Literatur</b>	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

<b>Code</b> ISWRE	<b>Name</b> Requirements Engineering	
<b>LP</b> 8	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> jedes 2. Sommersemester
<b>Format</b> Vorlesung 3 SWS + Übung 3SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung im Team	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Angewandte Informatik
<b>Sprache</b> Deutsch	<b>Lehrende</b> Barbara Paech	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	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)	
<b>Lerninhalte</b>	Methoden, Prozesse und Werkzeuge für Unternehmensmodellierung Prozessverbesserung in Unternehmen Anforderungserhebung und -verhandlung Requirements Management (Verbreitung, Prüfung und Aktualisierung von Anforderungen)	
<b>Teilnahmevoraussetzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden	
<b>Vergabe der LP und Modulendnote</b>	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.	
<b>Nuetzliche Literatur</b>	Wird jährlich aktualisiert	

## Scientific Visualization

<b>Code</b> ISV	<b>Name</b> Scientific Visualization	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every 3rd semester
<b>Format</b> Lecture 4 SWS + Exercise 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h on-campus program 15 h exam preparation 135 h independent study and exercises (possibly in groups)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Filip Sadlo	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	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.	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Introduction</li> <li>- Visualization Process</li> <li>- Data Sources and Representation</li> <li>- Interpolation and Filtering</li> <li>- Approaches for Visual Mapping</li> <li>- 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</li> <li>- Vector Field Visualization: Vector Calculus, Particle Tracing on Grids, Vector Field Topology, Vortex Visualization, Feature Extraction, Feature Tracking</li> <li>- Tensor Field Visualization: Glyphs, Hue-Balls and Lit-Tensors, Line-Based Visualization, Tensor Field Topology, Feature Extraction</li> </ul>	
<b>Teilnahmevoraussetzungen</b>	strongly recommended is: Computer Graphics (ICG) recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded oral or written exam. 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.	
<b>Nuetzliche Literatur</b>	C.D. Hansen, C.R. Johnson, The Visualization Handbook, 2005.	

## Software Evolution

<b>Code</b> ISWEvol	<b>Name</b> Software Evolution	
<b>LP</b> 3	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> unregelmäßig
<b>Format</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 90h; davon 30 h Vorlesung 35 Aufgabenbearbeitung und Aufarbeitung/Selbststudium 25 h Prüfungsvorbereitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
<b>Sprache</b> Deutsch	<b>Lehrende</b> Eckhart von Hahn	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>Die Studierenden können nach der Vorlesung: ein Software-Reengineering-Projekt fachlich planen und beurteilen, bei der Ersterstellung von Software die Evolutionsfähigkeit konzeptuell sicherstellen, ein Wartungskonzept für eine erstellte Software aufbauen.</p> <p>Sie kennen die Unterschiede und Herausforderungen der Software-Weiterentwicklung versus der Softwareneuentwicklung - und worauf die/der InformatikerIn hierbei achten muss, sowohl aus Sicht eines Softwareherstellers als auch aus der Sicht der NutzerInnen von Software, die klassischen Techniken der Softwaresanierung, die Typologie der Softwarewartung und das Management der Fehlerbehebung, die Relevanz der Thematik in der Praxis der industriellen Softwareerstellung.</p>	
<b>Lerninhalte</b>	<p>Dieses Modul vermittelt aus konzeptioneller Sicht die Grundlagen für ein erfolgreiches Lebenszyklusmanagement von Software nach ihrer Ersterstellung. Die Vorlesungsinhalte wurden aufgrund der 15-jährigen Erfahrung des Lehrenden in der Praxis zusammengestellt, auf Basis aktueller Forschung und Lehre.</p> <p>Einleitung Begriffsklärung, Grundlagen Softwareevolution Softwarewartung, Softwareerhaltung Software-Reengineering Evolution und Weiterentwicklung Management der Softwareevolution Zusammenfassung</p>	
<b>Teilnahme- voraus- setzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden	

<b>Vergabe der LP und Modulendnote</b>	Das Modul wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.
<b>Nuetzliche Literatur</b>	<p>Arnold, R. (Hrsg.): Software Reengineering. IEEE Computer Society Press, Los Alamitos 1993</p> <p>Fowler, M.: Refactoring ? Improving the Design of Existing Code. Addison-Wesley, Reading, Massachusetts, 1999</p> <p>von Hahn, E.: Werterhaltung von Software. DUV, Wiesbaden 2005</p> <p>Müller, B.: Reengineering. Eine Einführung. Teubner, Stuttgart 1997</p> <p>Sneed, H.M.; Hasitschka, M.; Teichmann, M.-T.: Software-Produktmanagement. Wartung und Weiterentwicklung bestehender Anwendungssysteme. dpunkt, Heidelberg 2005</p> <p>Smith, D.D.: Designing Maintainable Software. Springer, Heidelberg 1999</p>

## Software Ökonomie

<b>Code</b> ISWÖk	<b>Name</b> Software Ökonomie	
<b>LP</b> 3	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> unregelmäßig
<b>Format</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 90h; davon 30 h Vorlesung 35 Aufgabenbearbeitung und Aufarbeitung/Selbststudium 25 h Prüfungsvorbereitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
<b>Sprache</b> Deutsch	<b>Lehrende</b> Eckhart von Hahn	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>Die Studierenden können nach der Vorlesung grob den Preis und die Lizenzierung einer erstellten Software ermitteln, die Vermarktung von Software planen und anstoßen, grob die Bilanz sowie die Gewinn- und Verlustrechnung eines Softwareherstellers verstehen, den Wert einer Software mit seinen verschiedenen Komponenten beurteilen, aus Sicht des Herstellers sowie aus Sicht der Nutzer, Preisverhandlungen zu Softwareprojekten planen. Sie kennen die Grundzüge der Kosten- und Leistungsrechnung (soweit sie für die Softwareerstellung relevant ist), die unterschiedlichen Vertragsarten, die im Umfeld der Softwareerstellung zum Einsatz kommen, die wichtigsten Verhandlungsstrategien bei der Verhandlung von Softwareverträgen, rechtliche Aspekte im Bereich der IT-Kriminalität, die Relevanz der Vorlesungsthemen in der Praxis der industriellen Softwareerstellung.</p>	

<b>Lerninhalte</b>	<p>Dieses Modul vermittelt aus konzeptioneller Sicht die Grundlagen der Wirtschaftsinformatik, wie sie für die Softwareerstellung relevant sind. Die Vorlesungsinhalte wurden aufgrund der 15-jährigen Erfahrung des Dozenten in der Praxis zusammengestellt, auf Basis aktueller Forschung und Lehre.</p> <p>Einleitung  Begriffsklärung, Grundlagen  Softwareökonomie  Management von Softwareprojekten  Wertermittlung von Software  Bepreisung von Software  Software-Marketing  Verhandlungen und Verträge  Bilanzierung und Rechnungslegung  IT-Kriminalität  Schadensabwendung  Zusammenfassung</p>
<b>Teilnahmevoraussetzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden
<b>Vergabe der LP und Modulendnote</b>	Das Modul wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.
<b>Nuetzliche Literatur</b>	<p>Buxmann, P.; Diefenbach, H.; Hess, T.: Die Softwareindustrie. Ökonomische Prinzipien, Strategien, Perspektiven. Heidelberg, 2008</p> <p>Herzwurm, G.; Pietsch, W.: Management von IT-Produkten. Heidelberg, 2009</p> <p>Mertens, P. (Hrsg.): Lexikon der Wirtschaftsinformatik, 3. Auflage, Heidelberg 2001</p> <p>Versteegen, G.: Marketing in der IT-Branche. Heidelberg 2003</p> <p>Wöhe, G.: Einführung in die Allgemeine Betriebswirtschaftslehre. München 2010</p>

## Text Analytics

<b>Code</b> ITA	<b>Name</b> Text Analytics	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every 2nd winter semester
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h lecture 15 h preparation for exam 135 h self-study and working on assignments/projects (optionally in groups)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Michael Gertz	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>Students</p> <ul style="list-style-type: none"> <li>- can implement and apply different text analytics methods using open source NLP and machine learning frameworks</li> <li>- can describe different document and text representation models and can compute and analyze characteristic parameters of these models</li> <li>- know how to determine, apply, and interpret use-case specific document similarity measures and underlying ranking concepts</li> <li>- know the concepts and techniques underlying different text classification and clustering approaches</li> <li>- know different models for phrase extraction and text summarization and are able to apply respective models and concepts using NLP and machine learning frameworks</li> <li>- know the fundamental methods for the extraction of document outlines at different levels of granularity</li> <li>- are familiar with basic concepts of topic models and their application in different text analytics tasks</li> <li>- understand the principles of evaluating results of text analytics tasks</li> <li>- know the theoretical background of machine learning methods at sufficient depths to be able to choose parameters and adapt an algorithm to a given text analytics problem</li> <li>- are aware of ethical issues arising from applying text analytics in different domains</li> </ul>	

<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Text analytics in the context of Data Science</li> <li>- Open source text analytics, NLP, and machine learning frameworks</li> <li>- Fundamentals of NLP pipeline components</li> <li>- Document and text representation models</li> <li>- Document and text similarity metrics</li> <li>- Approaches, techniques and corpora for benchmarking text analytics tasks</li> <li>- Traditional and recent text classification and clustering approaches</li> <li>- Information extraction and topic detection approaches</li> <li>- Fundamentals of keyword and phrase extraction</li> <li>- Text summarization techniques</li> <li>- Generating document and text outlines</li> <li>- Ethical and legal aspects of text analytics methods</li> <li>- Text Analytics project management</li> </ul>
<b>Teilnahmevoraussetzungen</b>	Recommended are: solid knowledge of basic calculus, statistics, and linear algebra; good Python programming skills
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded exam. The note of this exam gives the note for this module. Details for this exam as well as the requirements for the assignment of credits will be given by the lecturer an the beginning of this course.
<b>Nuetzliche Literatur</b>	<p>The following textbooks and texts are useful but not required.</p> <ul style="list-style-type: none"> <li>- Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft)</li> <li>- Yoav Goldberg. A Primer on Neural Network Models for Natural Language Processing (2015)</li> <li>- Christopher D. Manning and Hinrich Schütze: Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: May 1999.</li> </ul> <p>Furthermore, during the course of this lecture, several papers covering topics discussed in class will be provided.</p>

## Visualisierung im Bereich Cultural Heritage

<b>Code</b> IVCH	<b>Name</b> Visualisierung im Bereich Cultural Heritage	
<b>LP</b> 2	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> unregelmäßig
<b>Format</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 60 h; davon 30 h Präsenzstudium, 30 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Angewandte Informatik
<b>Sprache</b> Deutsch	<b>Lehrende</b> Susanne Krömker	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	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).	
<b>Lerninhalte</b>	Weißlicht- und Time-of-flight-Scanner, Rekonstruktionen von Gefäßen und Gebäuden, 3D-Puzzle, Skelettierung, ethische Grundsätze	
<b>Teilnahmevoraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Computergraphik 1 (ICG1)	
<b>Vergabe der LP und Modulendnote</b>	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.	
<b>Nützliche Literatur</b>	Clive Orton: Mathematics in Archaeology. Cambridge, MA, Cambridge University Press, 1982 Katsushi Ikeuchi, Daisuke Miyazaki (editors): Digitally Archiving Cultural Objects. Springer, 2007	

## Volumenvisualisierung

<b>Code</b> IVV	<b>Name</b> Volumenvisualisierung	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every summer semester
<b>Format</b> Lecture 4 SWS + Exercise course 3 SWS	<b>Arbeitsaufwand</b> 240 h; davon 75 h Präsenzstudium 15 h Prüfungsvorbereitung 150 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik, M.Sc. Data and Computer Science,
<b>Sprache</b> English	<b>Lehrende</b> Jürgen Hesser	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<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>	
<b>Lerninhalte</b>	<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>	
<b>Teilnahmevoraussetzungen</b>	recommended are: Introduction into computer science I (IPI), programming course (IPK), algorithms & data structures (IAD);	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded written exam. 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.	
<b>Nuetzliche Literatur</b>	Engel et al.: Real-Time Volume Graphics <a href="http://www.real-time-volume-graphics.org">www.real-time-volume-graphics.org</a> , Schroeder et al.: VTK Textbook <a href="http://www.kitware.com/products/books/vtkbook.html">http://www.kitware.com/products/books/vtkbook.html</a>	

## Wissensmanagement und Entscheidungen im Software Engineering

<b>Code</b> ISWKM	<b>Name</b> Wissensmanagement und Entscheidungen im Software Engineering	
<b>LP</b> 3	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> jedes 2. Wintersemester
<b>Format</b> Vorlesung+Übung 2 SWS	<b>Arbeitsaufwand</b> 90 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
<b>Sprache</b> Deutsch	<b>Lehrende</b> Andrea Herrmann	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	Die Teilnehmer/innen kennen vertiefende Software Engineering Methoden, die Entscheidungen unterstützen bei Anforderungspriorisierung, Entwurf, Managemententscheidungen und Risikomanagement. Sie wissen, wie man im Arbeitsalltag Wissen verwaltet und haben eine Einführung in die Entscheidungstheorie erhalten.	
<b>Lerninhalte</b>	Wissensmanagement Ontologien Rationale Re-engineering learning organization Entscheidungen Management-Entscheidungen, Business Case Risikomanagement Anforderungspriorisierung Entscheidungen im Entwurf: ATAM, SAAM, CBAM Entscheidungstheorie Entscheiden unter Ungewissheit Mathematical Economics Entscheidung mit mehreren Parteien: Harvard-Konzept, Verhandlungen Spieltheorie Fehlentscheidungen/ Decision Traps/ Biases	
<b>Teilnahmevoraussetzungen</b>	empfohlen sind: Vorlesung und Übung Einführung in Software Engineering (ISW) oder vergleichbare Vorkenntnisse	
<b>Vergabe der LP und Modulendnote</b>	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.	
<b>Nützliche Literatur</b>	Raiffa, Howard; Richardson, John; Metcalfe, David: Negotiation analysis - the science and art of collaborative decision making, Belknap, Cambridge, 2002	

## 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):

Finite Elemente

Nichtlineare Optimierung

Numerische Optimierung bei Differentialgleichungen I

Uncertainty Quantification 1

- Grundmodul Statistik und Wahrscheinlichkeitsrechnung:

Wahrscheinlichkeitstheorie II

Statistik II

- Aufbaumodul Numerik und Optimierung (MM25):

Gemischte Finite Elemente

Parallele Löser für Finite Elemente

Numerische Optimierung bei Differentialgleichungen II

Uncertainty Quantification 2

- Spezialisierungsmodul Numerik und Optimierung (MM35):

Computational Fluid Dynamics

Fundamentals of Computational Environmental Physics

Mathematical Methods of Image and Pattern Analysis II

Konvexe Optimierung

Numerical Methods für Bayesian Inverse Problems

Numerical Methods for Maxwell's Equations

- Spezialisierungsmodul Statistik und Wahrscheinlichkeitsrechnung (MM36):

Statistische Analyse von maschinellen Lernalgorithmen

Aus den Ergänzungsmodulen:

- Berechenbarkeit und Komplexität I

- Berechenbarkeit und Komplexität II

## **4.5 Module aus dem M.Sc. Technische Informatik**

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