

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

**Universität Heidelberg**  
**Fakultät für Mathematik und Informatik**

**Fassung vom 16.10.2024 zur Prüfungsordnung vom 22.07.2010**  
**mit 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	Module aus der Informatik	17
	Advanced Machine Learning	18
	Algorithm Engineering	19
	Algorithms and Data Structures 2	21
	Applied Combinatorial Optimization	23
	Artificial Intelligence for Programming	26
	Complex Network Analysis	28
	Computational Geometry	30
	Computer Graphics	31
	Computer Vision	32
	Computer Games (Game Engine Design)	34
	Discrete Structures 1	36
	Discrete Structures 2	37
	Distributed and Parallel Algorithms	38
	Fortgeschrittenenpraktikum	40
	Fundamentals of Machine Learning	42
	Generative Neural Networks for the Sciences	43
	Geometric Modeling and Animation	44
	Hardware Aware Scientific Computing	46

Inverse Problems	48
IT Project Management	49
IT-Sicherheit 2	50
Knowledge Management and Decision-Making in Software Engineering	52
Machine Learning Essentials	53
Mining Massive Datasets	55
Natural Language Processing with Transformers	57
Object-Oriented Programming for Scientific Computing	59
Practical Geometry	60
Projektseminar Biomedizinische Bildanalyse	61
Randomisierte Algorithmen	63
Requirements Engineering	65
Scientific Visualization	66
Software Economics	67
Software Evolution	69
Time Series Analysis With Applications to Cognitive Science	71
Visualisierung im Bereich Cultural Heritage	73
Volume Visualization	74
4.3 Module aus dem B.Sc./M.Sc. Mathematik	75
4.4 Module aus dem M.Sc. Physik	76
4.5 Module aus dem M.Sc. Technische Informatik	77

# 1 Qualifikationsziele, Profil und Besonderheiten des Masterstudiengangs Angewandte Informatik

## 1.1 Präambel - Qualifikationsziele der Universität Heidelberg

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

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

## 1.2 Profil des Studiengangs

Der Masterstudiengang Angewandte Informatik wird von der Fakultät für Mathematik und Informatik getragen. Der Masterstudiengang ist forschungsorientiert. Er vertieft und verbreitert die Fachkenntnisse, befähigt zum selbstständigen wissenschaftlichen Arbeiten, legt die Voraussetzungen zur Weiterentwicklung des Faches und bereitet auf eine anspruchsvolle Berufstätigkeit oder eine Promotion vor. Absolventinnen und Absolventen sind qualifiziert für eigenverantwortliche und leitende Tätigkeiten. Der Masterstudiengang erlaubt eine sehr freie Gestaltung des Studiums um sowohl einen frühen Einstieg in forschungsnahe 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;</p> <p>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 M.Sc. Angewandte Informatik
<b>Sprache</b> Deutsch oder Englisch	<b>Lehrende</b> je nach Angebot	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<ul style="list-style-type: none"> <li>- Kenntnis von Techniken des wissenschaftlichen Schreibens (insbesondere auch Literaturrecherche) Fähigkeit, komplexe wissenschaftliche Literatur zu erschließen</li> <li>Erweiterte Fähigkeit, komplexe wissenschaftliche Literatur in einem Vortrag zu präsentieren,</li> <li>- Erweiterte Fähigkeit, zu Vorträgen zu diskutieren und Feedback zu geben,</li> <li>- Fähigkeit, ein kurze wissenschaftliche Ausarbeitung zu einem komplexen Thema zu erstellen.</li> </ul>	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Einführung in und Einübung von Techniken des wissenschaftlichen Schreibens,</li> <li>- Vertiefte Einübung der Erschließung und Präsentation wissenschaftlicher Literatur,</li> <li>- Fortgeschritteneres Informatikthema.</li> </ul>	
<b>Teilnahmevoraussetzungen</b>	empfohlen sind: Kenntnisse im Themengebiet des Seminars	
<b>Vergabe der LP und Modulendnote</b>	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.	
<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 Modulendnote</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>Teilnahme- voraus- setzungen</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. 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 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 Essentials (IMLE)

#### **Computergraphik und Visualisierung (CGV)**

Computational Geometry (ICGeo)  
Computer Graphics (ICG)  
Computer Games (Game Engine Design) (ICS)  
Computer Vision (ICV)  
Geometric Modeling and Animation (IGMA)  
Inverse Probleme (IIP)  
Practical Geometry (IPGeo)

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

### **Datenbanksysteme (DB)**

Complex Network Analysis (ICNA)

### **Optimierung**

Lineare Optimierung (MD3)  
Applied Combinatorial Optimization (IACO)

### **Parallele und Verteilte Systeme**

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

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

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

### **Technische Informatik**

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

### **Theoretische Informatik**

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

### **Wissenschaftliches Rechnen**

Advanced Machine Learning (IAML)  
Fundamentals of Machine Learning (IFML)  
Generative Neural Networks for the Sciences (IGNNS)  
Hardware Aware Scientific Computing (IHASC)  
Machine Learning Essentials (IMLE)  
Natural Language Processing with Transformers (INLPT)  
Numerik (MD1)  
Object-Oriented Programming for Scientific Computing (IOPSC)  
Time Series Analysis With Applications to Cognitive Science (ITSA-ACS)



## **4.2 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 Essentials“ 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>	<p>Students</p> <ul style="list-style-type: none"> <li>- get to know advanced machine learning methods that define the state-of-the-art and major research directions in the field,</li> <li>- understand when these methods are called for, what limitations of standard solutions they address, and how they are applied to real-world problems,</li> <li>- learn how to use Python-based machine learning software such as scikit-learn, theano and OpenGM.</li> </ul>	
<b>Lerninhalte</b>	<p>The lecture, along with its sibling “Fundamentals of Machine Learning“, offers an extended version of the one-semester course “Machine Learning“:</p> <p>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)</p>	
<b>Teilnahmevoraussetzungen</b>	recommended are: lecture “Fundamentals of Machine Learning“ or similar	
<b>Vergabe der LP und Modulendnote</b>	<p>The module is completed with a graded written examination. This examination is a report on a 90 h mini-research project. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations. Details will be given by the lecturer.</p>	
<b>Nuetzliche Literatur</b>	<p>David Barber: Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012</p> <p>Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, 2006</p>	

## 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</p> <ul style="list-style-type: none"> <li>- obtain a systematic understanding of algorithmic questions and solution approaches in the area of algorithm engineering,</li> <li>- are 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,</li> <li>- are able to select appropriate algorithms to come up with and implement efficient solutions, given a real-world problem,</li> <li>- know realistic machine models and applications, algorithm design, implementation techniques, experimental methodology and can interpret measurements.</li> </ul>	
<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>Teilnahme-voraus-setzungen</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 (IADS2)</p>	
<b>Vergabe der LP und Modulendnote</b>	<p>The module is completed with a graded oral examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.</p>	

<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>- advanced 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>Teilnahme-voraus-setzungen</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 examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.</p>	

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

## Applied Combinatorial Optimization

<b>Code</b> IACO	<b>Name</b> Applied Combinatorial Optimization	
<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  Cannot be combined with Optimization for Machine Learning.
<b>Sprache</b> English	<b>Lehrende</b> Bogdan Savchynskyy	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	The students - can analyze combinatorial optimization methods and estimate the area of their potential application; - can competently apply existing optimization algorithms and program packages; - know typical combinatorial optimization techniques and have a sufficient background for an independent literature search; - understand the basics of convex analysis, convex optimization, convex duality theory, (integer) linear programs and their geometry.	

<b>Lerninhalte</b>	<p>The course is devoted to combinatorial optimization, which includes but not limited to algorithms on graphs, integer linear programming, pseudo-boolean optimization, matroids and submodularity.</p> <p>A distinctive feature of this course is its motivation by machine learning applications, which shifts the optimization focus from attaining an optimal solution to a problem, to obtaining an accurate enough solution very fast. The reason for this shift is complexity of models used in modern artificial intelligence-related branches and the lesson they teach us: Better results can be easier attained by more accurate models rather than by more accurate optimization.</p> <p>To build an accurate problem model, the latter must be learnable. To be used in learning pipelines, combinatorial algorithms must be fast. To attain the best practical results, the algorithms must be accurate enough.</p> <p>Fast, accurate enough and learnable algorithms are three aspects we address in this lecture.</p> <ul style="list-style-type: none"> <li>- Combinatorial problems and their computational complexity: Overview</li> <li>- Linear and integer linear programs and their geometry: Convexity, polyhedra, LP relaxation.</li> <li>- Lifting of variables: Quadratic to linear problem transform, Sherali-Adams hierarchy</li> <li>- Lagrange duality: Subgradient, optimality conditions, relation to LP relaxation, reduced costs.</li> <li>- Systematic exact combinatorial methods: Branching and cutting.</li> <li>- Scalable dual techniques: Non-smooth first order methods, smoothing, primal-dual algorithm.</li> <li>- Greedy algorithms: (Sub-)Optimality, matroids.</li> <li>- Quadratic pseudo-boolean optimization: Algorithms, applications, submodularity.</li> <li>- Scalable primal heuristics: Greedy generation, local search and optimal recombination. Memetic algorithms.</li> <li>- Min-cost-flow: Problem subclasses, theoretical properties and practical algorithms.</li> <li>- Learning parameters of combinatorial problems from training data: Black-box differentiation and recent advances in the literature.</li> </ul>
<b>Teilnahmevoraussetzungen</b>	<p>recommended are: basic courses: Linear Algebra, Analysis (or, equivalently, Mathematics for computer science) and Algorithms and data structures.</p>
<b>Vergabe der LP und Modulendnote</b>	<p>The module is completed with a graded oral examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.</p>



<p><b>Nuetzliche Literatur</b></p>	<p>Savchynskyy, Bogdan. Discrete graphical models?an optimization perspective. Foundations and Trends® in Computer Graphics and Vision 11.3-4 (2019): 160-429.</p> <p>Boyd, Stephen P., and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004.</p> <p>Korte, Bernhard H. Combinatorial optimization. Berlin: Springer, 2011.</p> <p>Beck, Amir. First-order methods in optimization. Society for Industrial and Applied Mathematics, 2017.</p> <p>Bertsekas, Dimitri P. Nonlinear programming. Journal of the Operational Research Society 48.3 (1997): 334-334.</p> <p>Ahuja, Ravindra K., Thomas L. Magnanti, and James B. Orlin. Network flows. (1988).</p> <p>Papadimitriou, Christos H., and Kenneth Steiglitz. Combinatorial optimization: algorithms and complexity. Courier Corporation, 1998.</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>	Expected learning outcomes are: <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 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>	This module covers the following topics: <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>Teilnahme- voraus- setzungen</b>	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.	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	

<b>Nuetzliche Literatur</b>	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> Noah D. Goodman, Joshua B. Tenenbaum: Probabil-istic Models of Cognition (2nd ed.), 2016. Online: <a href="https://probmods.org/">https://probmods.org/</a> 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> 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>
-----------------------------	--

## 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 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. 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>	<p>The module is completed with a graded written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.</p>
<b>Nuetzliche Literatur</b>	<p>Albert-Laszlo Barabasi: Network Science, Cambridge University Press, 2016.  M.E.J. Newmann: Networks: An Introduction, Oxford University Press, 2010.  Vito Latora, Vincenzo Nicosia, Giovanni Russo: Complex Networks - Principles, Methods and Applications, Cambridge University Press, 2017.  David Easley, Jon Kleinberg: Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.  Stanley Wasserman, Katherine Faust: Social Network Analysis-Methods and Applications, Cambridge University Press, 1994.</p>

## 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> <ul style="list-style-type: none"> <li>- 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,</li> <li>- master the associated data structures for efficient storage and further processing and can calculate the complexity of the various algorithms.</li> </ul>	
<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: Algorithmen und Datenstrukturen (IAD)	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded oral examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
<b>Nuetzliche Literatur</b>	<p>Rolf Klein: Algorithmische Geometrie, Springer Verlag, 2005  Herbert Edelsbrunner: Geometry and Topology of Mesh Generation, Cambridge University Press, 2001  Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars:  Computational Geometry - Algorithms and Applications, 3rd edition, Springer, 2008  current publications</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>	<p>The students</p> <ul style="list-style-type: none"> <li>- understand fundamental and advanced concepts of computer graphics,</li> <li>- understand the mathematical fundamentals, data structures, and implementation aspects,</li> <li>- get to know raster graphics, geometric transforms, color perception and color models, and basics of geometric modeling,</li> <li>- are able to apply these concepts to real-world problems using existing software packages, and develop small programs using OpenGL 4.</li> </ul>	
<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>Teilnahme- voraus- setzungen</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 examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
<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>	

## Computer Vision

<b>Code</b> ICV	<b>Name</b> Computer Vision	
<b>LP</b> 6	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every semester
<b>Format</b> Lecture 2 SWS + Exercise 2 SWS	<b>Arbeitsaufwand</b> 180 h; thereof 30 h lectures 30 h exercises 20 h revision and home exercise 70 h programming a mini research project 30 h preparation of final report	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing  renaming of Computer Vision: 3D Reconstruction
<b>Sprache</b> English	<b>Lehrende</b> Carsten Rother	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>The students</p> <ul style="list-style-type: none"> <li>- understand and implement the principles behind estimating 3D Point-Clouds and Motion from two or more images. They are able to apply this knowledge to new tasks in the field of 3D reconstruction.</li> <li>- understand the principles of image processing and image formation. This can be utilized to design an algorithm for camera calibration.</li> <li>- have studied various techniques for fast object recognition. This can be used to build an object recognition system for e.g. autonomous driving.</li> <li>- understand different approaches for object tracking.</li> <li>- have studied methods for conditional image generation. This can be used to build an image generation technique in a new domain, e.g. fashion design.</li> <li>- understand and implement methods that combine machine learning-based methods with classical computer vision-based techniques.</li> <li>- have studied various state-of-the-art computer vision systems and approaches, and are then able to evaluate and classify new systems and approaches.</li> </ul>	
<b>Lerninhalte</b>	<p>This lecture covers a broad range of areas in computer vision: Image Processing, 3D Reconstruction, Object Tracking, Image Understanding, and Image Generation. For instance, we will discuss the underlying techniques and associated theory to recover a 3D scene from a set of photographs. A focus of the lecture is to investigate techniques from deep learning, e.g. vision transformers, traditional approaches, e.g. RANSAC, and mixtures of the two, e.g. Differentiable RANSAC. We also introduce the necessary background knowledge, e.g. basic Deep Learning, Image Formation Models, Camera Models, Kalmann Filters, Diffusion Models, etc.</p>	
<b>Teilnahmevoraussetzungen</b>	<p>recommended is a basic machine learning background (e.g. Fundamentals of Machine Learning, Advanced Machine Learning or equivalent)</p>	



<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded examination. This examination is either a graded final report (about 10 pages) or a final report (about 5 pages) together with an oral examination. The grade of this examination gives the grade for this module. Details for this examination as well as the requirements for the assignment of credits will be given by the lecturer an the beginning of this course.
<b>Nuetzliche Literatur</b>	

## Computer Games (Game Engine Design)

<b>Code</b> ICS	<b>Name</b> Computer Games (Game Engine Design)	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every summer semester
<b>Format</b> Lecture 3 SWS + Exercise 3 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 75 h lectures and tutorials 15 h exam preparations 150 h self-study and exercises	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Jürgen Hesser	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>The students</p> <ul style="list-style-type: none"> <li>- understand the game engine concepts, the decision for specialized class structures, support tools, and the typical architectural elements and are able to apply these concepts in developing an own game engine,</li> <li>- are able to apply and further develop methods for visualizing 3D scenes, perform collision detection and hereby are able to identify the appropriate algorithms,</li> <li>- have the capability to develop animation methods with different levels of complexity and are able to assess which method to take under the trade-off between performance and quality,</li> <li>- will be able to find and apply appropriate techniques for path planning, to improve the found paths to be more realistic,</li> <li>- are able to identify the different concepts of AI in games and develop and apply these techniques for own games.</li> </ul> <p>In the exercises, they apply the theoretical concepts and program applications in order to see how to translate concepts into code.</p>	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Overview of the structure and the components of computer games</li> <li>- Architecture of Game Engines</li> <li>- Elements of the Graphics Subengine</li> <li>- Algorithms for Collision Detection</li> <li>- Animation techniques and physics</li> <li>- Path planning and AI</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 oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	

<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 (im BSc Mathematik 1+2)
<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 graphs 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>Teilnahme- voraus- setzungen</b>	recommended are: Einführung in die Praktische Informatik (IPI), Mathematik für Informatiker 1 (IMI1) or Lineare Algebra 1 (MA4), Mathematik für Informatiker 2 (IMI2) or Analysis 1 (MA1)	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
<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>	Students - understand several advanced graph parameters and the central theorems in these areas, - can solve problems involving discussed topics, - can reprove the central considered results.	
<b>Lerninhalte</b>	- Probabilistic Methods - Extremal graph theory - Expander graphs - Quasirandom graphs - Further advanced topics	
<b>Teilnahmevoraussetzungen</b>	recommended: Discrete Structures 1	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
<b>Nuetzliche Literatur</b>	Reinhard Diestel Graph Theory, 5th edition, Springer, 2016/17 Douglas West, Introduction to Graph Theory, Pearson, 2011. J.A. Bondy and U.S.R. Murty, Graph Theory, Springer, 2008. Bernhard Korte and Jens Vygen, Combinatorial Optimization, 6th edition, 2018.	

## Distributed and Parallel Algorithms

<b>Code</b> IDPA	<b>Name</b> Distributed and Parallel Algorithms	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every 3rd to 4th semester
<b>Format</b> 4 SWS lecture 2 SWS tutorial, homework assignments	<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</p> <ul style="list-style-type: none"> <li>- understand fundamental theoretical and practical concepts of advanced parallel 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 where parallel or distributed algorithms are used,</li> <li>- are able to analyse new distributed and parallel algorithms as well as analysing their running time,</li> <li>- and select appropriate algorithms for parallel and distributed applications,</li> <li>- are able to apply parallel and distributed algorithms and data structures to real-world problems,</li> <li>- can objectively assess the quality of the results.</li> </ul>	
<b>Lerninhalte</b>	<p>Introduction to distributed and parallel algorithms, PRAM model, design and analysis of parallel and distributed algorithms, isoefficiency, UMA vs. NUMA, memory consistency for shared-memory, communication models (with and without network, fully interconnected with half duplex or full duplex, BSP), critical path lengths, parallel associative operations, reduction operations, matrix multiplication, broadcast operations, MPI basic toolbox, ranking, parallel sorting (multiway merge, quick sort, sample sort), prefix sums, all-to-all communication, map-reduce, list ranking, parallel graph algorithms (minimum spanning trees, connected components, shortest paths, graph partitioning), process mapping, communication-free parallel graph generation, parallel sampling algorithms.</p>	
<b>Teilnahmevoraussetzungen</b>	<p>recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Lineare Algebra 1</p>	
<b>Vergabe der LP und Modulendnote</b>	<p>The module is completed with a graded oral examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.</p>	

<b>Nuetzliche Literatur</b>	Sanders, Mehlhorn, Dietzfelbringer, Dementiev. Sequential and Parallel Algorithms and Data Structures. 2019. Kumar, Grama, Gupta, Karypis. Introduction to Parallel Computing. Design and Analysis of Algorithms. 1994 Leighton. Introduction to Parallel Algorithms and Architectures. 1992 Jaja. An Introduction to Parallel Algorithms. 1992
---------------------------------	--

## Fortgeschrittenenpraktikum

<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
<b>Sprache</b> Deutsch oder Englisch	<b>Lehrende</b> je nach Angebot	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>- erlangen vertiefende Problemlösungskompetenz für komplexe Entwurfs- und Implementierungsaufgaben,</li> <li>- können Problemanalyse- und Beschreibungstechniken klar darstellen, differenzieren und anwenden,</li> <li>- vertiefen Programmierkenntnisse in der jeweiligen für das Projekt erforderlichen Programmiersprache,</li> <li>- sind in der Lage, das Projekt mit Hilfe einer Softwareentwicklungsumgebung durchzuführen.</li> </ul> <p>Zusätzlich werden die projekttypischen Kompetenzen vertieft, insbesondere das Arbeiten im Team (von bis zu drei Studierenden):</p> <ul style="list-style-type: none"> <li>- Durchführung und Evaluation von Projekten und ihrer Phasenstruktur,</li> <li>- Planung und Durchführung von Projekt- und Teamarbeit.</li> </ul> <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:</p> <ul style="list-style-type: none"> <li>- Vertiefung in die Projektarbeit,</li> <li>- Eigenständige Entwicklung von komplexer Software und deren Dokumentation.</li> </ul>	
<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>Nuetzliche 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 Essentials“ 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>	<p>Students</p> <ul style="list-style-type: none"> <li>- understand fundamental concepts of machine learning (features vs. response, unsupervised vs. supervised training, regression vs. classification etc.),</li> <li>- get to know established learning methods and algorithms,</li> <li>- are able to apply them to real-world problems, and can objectively assess the quality of the results,</li> <li>- learn how to use Python-based machine learning software such as scikit-learn.</li> </ul>	
<b>Lerninhalte</b>	<p>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:</p> <ul style="list-style-type: none"> <li>- Classification (nearest neighbor rules, linear and quadratic discriminant analysis, logistic regression, classical and randomized decision trees, support vector machines, ensemble methods)</li> <li>- Regression (linear and non-linear least squares, regularized and sparse regression, robust regression)</li> <li>- Unsupervised learning (hierarchical clustering, k-means algorithm, Gaussian mixture models and expectation maximization, principal component analysis, non-linear dimension reduction)</li> <li>- Evaluation (risk minimization, model selection, cross-validation)</li> </ul>	
<b>Teilnahme- voraus- setzungen</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 examination. This examination is a report on a 90 h mini-research project. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations. Details will be given by the lecturer.	
<b>Nuetzliche Literatur</b>	Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning (2nd edition), Springer, 2009	

## Generative Neural Networks for the Sciences

<b>Code</b> IGNNS	<b>Name</b> Generative Neural Networks for the Sciences	
<b>LP</b>	<b>Dauer</b>	<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> 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 a broad range of generative neural network design and learning methods, with an emphasis on solving problems in the sciences, - understand the strengths and limitations of these methods, can apply them to real-world problems and objectively assess the quality of the results, - familiarize themselves with important open-source implementations of these methods.	
<b>Lerninhalte</b>	- Types of generative neural networks: normalizing flows, diffusion models, (variational) autoencoders, recurrent networks, transformers - Techniques: simulation-based inference, hierarchical models, physics-informed neural networks, symbolic regression, causal discovery - Quality diagnostics: predictive accuracy, probabilistic calibration, re-simulation error, disentanglement scores, generalization ability, and pitfalls of those diagnostics - Applications: design of efficient surrogates for classical models, Bayesian inference for inverse problems, analysis of dynamic systems, with examples from physics, medicine, engineering, cognitive science, and others	
<b>Teilnahme- voraus- setzungen</b>	recommended: basic knowledge of deep learning and statistics	
<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>	Kevin Murphy. Probabilistic Machine Learning: Advanced Topics (2023)	

## 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> <ul style="list-style-type: none"> <li>- know the mathematical foundations of geometric modeling,</li> <li>- know the mathematical and physical foundations of computer animation,</li> <li>- know the algorithms and implementation aspects,</li> <li>- are familiar with the basics of animated movies,</li> <li>- are able to apply existing tools for geometric modeling and animation.</li> </ul>	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Introduction to curves</li> <li>- Interpolating curves</li> <li>- Bézier curves</li> <li>- B-Splines</li> <li>- Rational curves</li> <li>- Introduction to surfaces</li> <li>- Tensor product surfaces</li> <li>- Transfinite surfaces and extrusion</li> <li>- Subdivision</li> <li>- Subdivision surfaces</li> <li>- Animation and simulation</li> <li>- Rigid body kinematics</li> <li>- Particle systems</li> <li>- Mass-spring models</li> <li>- Cloth modeling</li> <li>- Numerical methods for differential equations</li> <li>- Collision detection and handling</li> <li>- Fluid simulation and natural phenomena</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 oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	

<b>Nuetzliche Literatur</b>	<p>Curves and Surfaces for CAGD - A Practical Guide, G. Farin, Morgan Kaufmann, 2002</p> <p>Computer Animation - Algorithms and Techniques, R. Parent, Morgan Kaufmann, 2002</p> <p>3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, D. Eberly, Morgan Kaufmann, 2000</p> <p>Graphische Datenverarbeitung I, J. Encarnacao, W. Straßer, R. Klein, 4. Auflage, Oldenbourg 1996</p> <p>Advanced Animation and Rendering Techniques, A. Watt, M. Watt, Addison-Wesley, 1992</p> <p>Grundlagen der geometrischen Datenverarbeitung, J. Hoschek, D. Lasser, Teubner 1992</p> <p>Numerical Recipes - The Art of Scientific Computing, W.H. Press, P. Flannery, S.A. Teukolsky, W.T. Vetterling, Cambridge University Press, 1986</p>
---------------------------------	---

## 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>	<p>Students</p> <ul style="list-style-type: none"> <li>- are familiar with different forms of parallelism in modern computer architectures,</li> <li>- can exploit this parallelism selecting an appropriate programming model,</li> <li>- are familiar with modelling of parallelism and know fundamental parallel algorithms from scientific computing.</li> </ul>	
<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 examination. The final grade of the module is determined by the grade of the examination. Details for this examination as well as the requirements for the assignment of credits will be given by the lecturer an the beginning of this course.
<b>Nuetzliche Literatur</b>	Frédéric Magoules, François-Xavier Roux, Guillaume Houzeaux: Parallel Scientific Computing, Wiley, 2016, doi: 10.1002/9781118761687

## Inverse Problems

<b>Code</b> IIP	<b>Name</b> Inverse Problems	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every summer semester
<b>Format</b> Lecture 2 SWS + Exercise course 2 SWS + Homework	<b>Arbeitsaufwand</b> 240 h; thereof 60 h lectures and tutorials 15 h exam preparations 165 h self-study and exercises / homework	<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>Students</p> <ul style="list-style-type: none"> <li>- understand the mathematical properties of inverse problems and are able to demonstrate and show why these problems are difficult to solve,</li> <li>- learn principles of how to solve both deterministic and stochastic problems,</li> <li>- they are able to identify problem settings which request specific deterministic or stochastic approaches and the regularization methods therein,</li> <li>- are able to select an appropriate regularization parameter strategy and understand their differences in particular,</li> <li>- understand how to formulate and apply compressed sensing and deep learning for inverse problems, all principles are presented in selected areas in parameter estimation,</li> <li>- gain the competence in solving complex problems that cannot be dealt with classical techniques,</li> <li>- will be able to adequately evaluate complex experimental measurements.</li> </ul>	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Definition of ill-posedness</li> <li>- Deterministic inverse problems, regularization techniques</li> <li>- Tikhonov regularization, data and model resolution matrix, pseudo-inverses</li> <li>- Stochastic inverse problems and Bayes theorem</li> <li>- Regularization parameter selection</li> <li>- Compressed sensing</li> <li>- Deep Learning for Inverse Problems</li> </ul>	
<b>Teilnahmevoraussetzungen</b>	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Numerische Mathematik	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
<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 Project Management

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

## IT-Sicherheit 2

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

<b>Vergabe der LP und Modulendnote</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>	

## Knowledge Management and Decision-Making in Software Engineering

<b>Code</b> ISWKM	<b>Name</b> Knowledge Management and Decision-Making in Software Engineering	
<b>LP</b> 3	<b>Dauer</b> one semester	<b>Angebotsturnus</b> every second winter semester
<b>Format</b> lecture + exercise 2 SWS	<b>Arbeitsaufwand</b> 90 h; thereof: 30 h lecture + exercise 15 h preparation for exam 45 h self-study and homework (optionally in groups)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
<b>Sprache</b> English	<b>Lehrende</b> Andrea Herrmann	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	The students - know advanced software engineering techniques which support decision-making during requirements prioritization, design, management decisions and risk management, - know how to manage knowledge in every day work life and got an introduction into decision theory.	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Knowledge management</li> <li>- Ontologies and Grounded Theory</li> <li>- Reverse engineering, code metrics</li> <li>- Learning organization</li> <li>- Storytelling</li> <li>- Decision-making and decision theory</li> <li>- Management decisions, business case</li> <li>- Risk management</li> <li>- Requirements prioritization</li> <li>- Decision-making in design: ATAM, SAAM, CBAM</li> <li>- Decision-making under uncertainty</li> <li>- Mathematical Economics</li> <li>- Decision-making with several parties: Harvard concept, negotiations, Game Theory</li> <li>- Decision Traps and Biases</li> <li>- Ethical decisions and machine ethics</li> </ul>	
<b>Teilnahme- voraus- setzungen</b>	recommended are: Einführung in Software Engineering (module ISW) or comparable competences	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded (oral or written) examination. The grade of the module is the grade of the examination. Prerequisite for the participation in the exam are 50% of the points for the homework.	
<b>Nuetzliche Literatur</b>	Raiffa, Howard; Richardson, John; Metcalfe, David: Negotiation analysis - the science and art of collaborative decision making, Belknap, Cambridge, 2002 or 2007	

## Machine Learning Essentials

<b>Code</b> IMLE	<b>Name</b> Machine Learning Essentials	
<b>LP</b> 8	<b>Dauer</b> one semester	<b>Angebotsturnus</b> in (irregular) alternation with “Fundamentals of Machine Learning“ and “Advanced 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> This is the retitled “Machine Learning“ module, 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>	<p>The students</p> <ul style="list-style-type: none"> <li>- understand a broad range of machine learning concepts, get to know established and advanced learning methods and algorithms,</li> <li>- are able to apply them to real-world problems, and can objectively assess the quality of the results.</li> <li>- learn how to use Python-based machine learning software such as scikit-learn.</li> </ul>	
<b>Lerninhalte</b>	<p>This lecture is a compact version of the two-semester course “Fundamentals of Machine Learning“ and “Advanced Machine Learning“:</p> <p>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</p>	
<b>Teilnahmevoraussetzungen</b>	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	

<b>Vergabe der LP und Modulendnote</b>	<p>This is the retitled “Machine Learning“ module.</p> <p>The module is completed with a graded written examination. This examination is a report on a 90 h mini-research project. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations. Details will be given by the lecturer.</p>
<b>Nuetzliche Literatur</b>	<p>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</p>

## 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>	<p>The students</p> <ul style="list-style-type: none"> <li>- know selected approaches and programming paradigms of parallel data processing,</li> <li>- know how to use tools for parallel data processing (among others Apache Hadoop and Spark),</li> <li>- are familiar with application domains of big data analysis,</li> <li>- know methods of parallel pre-processing of data,</li> <li>- know methods like classification, regression, clustering and their parallel implementations,</li> <li>- know about 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>Teilnahme- voraus- setzungen</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 examination. The final grade of the module is determined by the grade of the examination. Details for this examination as well as the requirements for the assignment of credits will be given by the lecturer an the beginning of this course.	

<b>Nuetzliche Literatur</b>	<p>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>)</p> <p>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>)</p> <p>Ron Bekkerman, Misha Bilenko, John Langford, Scaling Up Machine Learning, Cambridge University Press, 2012</p> <p>Jiawei Han, Micheline Kamber, Jian Pei, Data Mining: Concepts and Techniques, Morgan Kaufmann, (third edition), 2012</p> <p>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>)</p>
-----------------------------	--



## Natural Language Processing with Transformers

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

<b>Teilnahmevoraussetzungen</b>	<p>Recommended courses: Data Science for Text Analytics (IDSTA), Foundations of Machine Learning (IML)</p> <p>Recommended background: solid knowledge of basic calculus, statistics, and linear algebra; good Python programming skills; familiarity with frameworks such as Huggingface, Google Colab, and cloud-based services, in particular vector databases</p>
<b>Vergabe der LP und Modulendnote</b>	<p>Assignments (40%) and Programming Project (60%); about 4-5 assignments focusing on the material learned in class on a conceptual and formal level; group project in which 3-4 students develop a prototypical transformer-based application, including design and evaluation, a written project documentation as well as the code need to be submitted at the end of the class, clearly indicating which part of the project each student is responsible for. Both assignments and project must be at least satisfactory (4,0) in order to pass the class.</p>
<b>Nuetzliche Literatur</b>	<p>For the different topics, several research papers will be provided to students via the Moodle platform. The following textbooks are useful but not required:</p> <p>Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Natural Language Processing with Transformers, 2022 (revised edition)</p> <p>Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft)</p> <p>Furthermore, during the course of this lecture, several papers covering topics discussed in class will be provided.</p>

## Object-Oriented Programming for Scientific Computing

<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> irregularly
<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> varying	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	<p>The students</p> <ul style="list-style-type: none"> <li>- are proficient in the programming language C++,</li> <li>- can assess the performance of different programming techniques,</li> <li>- know template programming techniques, and can use the Standard Template Library (STL),</li> <li>- can apply their new skills to solve selected problems of Scientific Computing.</li> </ul>	
<b>Lerninhalte</b>	<p>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:</p> <ul style="list-style-type: none"> <li>- Class concept,</li> <li>- Dynamic memory allocation,</li> <li>- Exception handling,</li> <li>- Resource allocation and initialization,</li> <li>- Constness,</li> <li>- Static versus dynamic polymorphism,</li> <li>- Traits and Policies,</li> <li>- Standard Template Library,</li> <li>- Template Metaprogramming,</li> <li>- Parallel programming techniques.</li> </ul>	
<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 examination. The final grade of the module is determined by the grade of the examination. The lecture will give the requirements for the assignment of credits.	
<b>Nuetzliche Literatur</b>		

## Practical Geometry

<b>Code</b> IPGeo	<b>Name</b> Practical Geometry	
<b>LP</b> 4	<b>Dauer</b> one semester	<b>Angebotsturnus</b> irregularly
<b>Format</b> Lecture 2 SWS + Exercise course 1 SWS	<b>Arbeitsaufwand</b> 120h; thereof 45 h lecture 60 h self-study and working on assignments 15h preparation for exam	<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</p> <ul style="list-style-type: none"> <li>- understand basic geometric concepts for data analysis as well as efficient point search and further processing of measurement data</li> <li>- confidently handle projections and descriptions beyond the three-dimensional world of experience,</li> <li>- calculate geometric invariants, distances, curvatures from measurement data, reconstructed and generated surfaces.</li> </ul>	
<b>Lerninhalte</b>	<p>Basic areas of geometry with relevance in computer graphics, image processing, pattern recognition, computer vision and geometric modeling.</p> <p>(i) Analytic geometry: operations on vector spaces with appropriate coordinates and mappings (affine mappings, collinearities), geometric fitting of point clouds to linear structures or planes from error-prone measurement data</p> <p>(ii) Projective geometry: central projection and inverse reconstruction of 3D objects from planar images (computer vision, geodesy), differences between B-spline curves and surfaces and the class of NURBS, freeform geometries in CAD systems</p> <p>(iii) Differential geometry: parameter representations in geometric data processing, implicit representations (level sets), estimation of invariants from discrete data (triangulations, point clouds).</p>	
<b>Teilnahmevoraussetzungen</b>	recommended are: linear algebra, computational geometry and any programming language (e.g. C/C++/Pascal/python)	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded oral examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
<b>Nuetzliche Literatur</b>	Geometrie für Informatiker, Script TU Vienna 2004, Helmut Pottmann, current publications	

## 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> 1+1
<b>Lernziele</b>	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>- erlangen vertiefte Kenntnisse und Fähigkeiten im Gebiet Biomedizinische Bildanalyse,</li> <li>- lernen fortgeschrittene Methoden und Algorithmen zur automatischen Analyse biomedizinischer Bilder,</li> <li>- lernen wie man Algorithmen und Software für automatische Bildanalyse entwickelt</li> </ul> <p>erweitern ihre Fähigkeiten Projektergebnisse mündlich zu präsentieren und schriftlich zu dokumentieren,</p> <ul style="list-style-type: none"> <li>- erweitern ihre Fähigkeiten zur Teamarbeit und zur Strukturierung von Projekten.</li> </ul>	
<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 Seminarteil (Einarbeitung in die relevante Literatur, Erarbeitung der theoretischen Grundlagen, Vortragspräsentation) und einem Projektteil (Spezifikation eines Softwaresystems, Entwurf von Algorithmen und Implementierung von Bildanalyseverfahren, Test und Evaluierung der Verfahren, Präsentation der Ergebnisse).</p>	
<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>	<p>Bekanntgabe in der Lehrveranstaltung</p>

## Randomisierte Algorithmen

<b>Code</b> IRA	<b>Name</b> Randomisierte Algorithmen	
<b>LP</b> 6	<b>Dauer</b> ein Semester	<b>Angebotsturnus</b> mind. 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>	<p>Auf der Grundlage der behandelten Anwendungsbeispiele aus verschiedenen Teilgebieten der Informatik können die Studierenden die probabilistische Betrachtungs- und Vorgehensweise anwenden</p> <ul style="list-style-type: none"> <li>- bei der Konstruktion und Analyse von probabilistischen und deterministischen Algorithmen,</li> <li>- auf kombinatorische Fragestellungen,</li> <li>- um spieltheoretische Situationen zu analysieren,</li> <li>- auf kryptographische Fragestellungen.</li> </ul>	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Elementare Wahrscheinlichkeitsrechnung,</li> <li>- Das Tenure-Spiel,</li> <li>- Derandomisierungstechniken,</li> <li>- Die probabilistische Methode,</li> <li>- Byzantinische Übereinkunft,</li> <li>- Stabile Heiraten und der Gale-Shapley-Algorithmus,</li> <li>- Das Minimax-Prinzip von Yao,</li> <li>- Komplexitätsanalyse des randomisierten Sortierens,</li> <li>- Randomisierte Fehlersuche und -korrektur,</li> <li>- Das Local-Lemma von Lovasz,</li> <li>- PAC-Lernen und VC-Dimension,</li> <li>- Wahrscheinlichkeitsverstärkung und Fehlerschranken,</li> <li>- Lokale Suche für k-SAT,</li> <li>- Kryptographische Protokolle.</li> </ul>	
<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 "Lerninhalte" 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>Teilnahme- voraus- setzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie im 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>	<p>The students</p> <ul style="list-style-type: none"> <li>- understand fundamental and advanced concepts of scientific visualization,</li> <li>- understand the mathematical fundamentals, data structures, and implementation aspects</li> <li>- get to know schemes for interpolation and integration, mapping for scalar, vector, and tensor fields, and derived approaches,</li> <li>- understand approaches for direct and indirect volume rendering, feature extraction, and topology-based analysis,</li> <li>- are able to apply these concepts to real-world problems using existing software packages, and develop small programs using visualization libraries.</li> </ul>	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <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>Teilnahme-voraus-setzungen</b>	<p>strongly recommended is: Computer Graphics (ICG) recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)</p>	
<b>Vergabe der LP und Modulendnote</b>	<p>The module is completed with a graded oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.</p>	
<b>Nuetzliche Literatur</b>	<p>C.D. Hansen, C.R. Johnson, The Visualization Handbook, 2005.</p>	

## Software Economics

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

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

## Software Evolution

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

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

## Time Series Analysis With Applications to Cognitive Science

<b>Code</b> ITSA-ACS	<b>Name</b> Time Series Analysis With Applications to Cognitive Science	
<b>LP</b>	<b>Dauer</b>	<b>Angebotsturnus</b> every summersemester
<b>Format</b> Lecture 4 SWS + Exercise course 2 SWS	<b>Arbeitsaufwand</b> 240h, thereof 60h lecture, 30h exercises, 126h self-study and working on assignments (optionally in groups), 24h exam preparation	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
<b>Sprache</b> English	<b>Lehrende</b> Georgia Koppe	<b>Prüfungsschema</b> 1+1
<b>Lernziele</b>	Students are familiarized with fundamental concepts of time series analysis, understand a variety of different time series models, and learn appropriate statistical inference frameworks. Students learn which models are suitable for a given problem, how to assess model performance, and how to select from a set of different models. The acquired knowledge enables students to generalize problem settings to new real world data sets, select or develop suitable statistical time series models for data analysis, and self-implement these models into code.	
<b>Lerninhalte</b>	Fundamental concepts in time series analysis, data preprocessing and visualization, linear regression, simple autoregressive models for stochastic processes (normal, Bernoulli, Poisson), model assessment and selection, cognitive computational models (discounted decision making, sequential sampling, reinforcement learning models), active learning with cognitive models, latent variable models, Hidden-Markov-models, state space models for stochastic processes (normal, Poisson), discrete-time nonlinear dynamical systems models (variants of recurrent neural network models and inference schemes)s	
<b>Teilnahme- voraus- setzungen</b>	recommended prior knowledge in basic calculus, statistics, and linear algebra	
<b>Vergabe der LP und Modulendnote</b>	The module is completed with a graded oral or written examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	

<p><b>Nuetzliche Literatur</b></p>	<p>Bishop, C. M., &amp; Nasrabadi, N. M. (2006). Pattern recognition and machine learning. New York: Springer.</p> <p>Durstewitz, D. (2017). Advanced data analysis in neuroscience. Bernstein Series in Computational Neuroscience. Cham: Springer.</p> <p>Hastie, T., Tibshirani, R., Friedman, J. H., &amp; Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction. New York: Springer.</p> <p>Murphy, K. P. (2022). Probabilistic machine learning: an introduction. MIT press.</p> <p>Shumway, R. H., Stoffer, D. S. (2017). Time series analysis and its applications: With R examples. Springer.</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, - 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), - 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	

## Volume Visualization

<b>Code</b> IVV	<b>Name</b> Volume Visualization	
<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; thereof 75 h lecture 15 h preparation for exam 150 h self-study and working on assignments/projects (optionally in groups)	<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</p> <ul style="list-style-type: none"> <li>- learn to understand how to use techniques of volume visualization to render complex scientific data, this consists of the representation of data by surface or volume elements, the conversion of different representations and techniques of interpolation,</li> <li>- 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.</li> </ul>	
<b>Lerninhalte</b>	<ul style="list-style-type: none"> <li>- Introduction of the visualization of scientific data of natural sciences and bio-sciences</li> <li>- Discrete and continuous representation of data and methods of interpolation</li> <li>- Methods of conversion between surface and volume representations and their efficient realizations</li> <li>- Theory of volume rendering and their different realizations</li> <li>- Acceleration and parallelization of volume rendering</li> <li>- Programming technique: GPU-programming</li> </ul>	
<b>Teilnahme- voraus- setzungen</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 examination. The final grade of the module is determined by the grade of the examination. The requirements for the assignment of credits follows the regulations in section modalities for examinations.	
<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>	

### 4.3 Module aus dem B.Sc./M.Sc. Mathematik

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

- Wahrscheinlichkeitstheorie (MC4)
- Numerik (MD1)
- Statistik (MD2)
- Grundlagen der Optimierung (MD3)

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

- Grundmodul Numerik und Optimierung (MM15):

Nichtlineare Optimierung

Uncertainty Quantification 1

- Spezialisierungsmodul Numerik und Optimierung (MM35):

Computational Fluid Dynamics

Fundamentals of Computational Environmental Physics

Mathematical Methods of Image and Pattern Analysis II

Aus den Ergänzungsmodulen:

- Berechenbarkeit und Komplexität I
- Berechenbarkeit und Komplexität II

## 4.4 Module aus dem M.Sc. Physik

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

## **4.5 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.