Module Handbook

Master Course of Studies "Data and Computer Science"

Ruprecht-Karls-Universität Heidelberg
Fakultät für Mathematik und Informatik

Version as of 09.02.2022, corresponding to examination regulations of 29.09.2021

Form of study: full time

Type of study: consecutive

Regular period of study: 4 semesters

Number of credit points to gain in this study: 120

Location of study: Heidelberg

Number of places: Unlimited

Fee: According to general regulations of Heidelberg University

Contents

1 Qualification objectives, profile, and particularities of the degree programme

1.1 Preamble - Qualification objectives of Heidelberg University

In keeping with Heidelberg University's mission statement and constitution, degree programmes are designed to provide a comprehensive academic education, incorporating subject-specific, cross-disciplinary, and career-related objectives that prepare students for their future professional careers. The resulting skills profile is a valid qualification profile that is included in the module handbooks for all university disciplines and is implemented in each degree programme?s specific qualification objectives, curricula, and modules:

- Development of subject-specific skills, with a particular emphasis on research;
- Development of the skills required for trans-disciplinary dialogue;
- Development of practical problem-solving skills;
- Development of personal and social skills;
- Promotion of students' willingness to assume social responsibility on the basis of the skills acquired.

1.2 Profile of the degree programme

The Master's program in Data and Computer Science is operated by the Faculty of Mathematics and Computer Science. The Master's program is research-oriented. It deepens and broadens the expertise, enables independent scientific work, lays the foundations for further development of the subject, and prepares students for a demanding professional career or a doctorate. Graduates are qualified for responsible and leadership activities.

The Master's program focuses on Data Science and Computer Science, thereby bridging the emerging field of Data Science with a well-established field. Particular interest in Data Science is put on subject areas such as machine learning, visual computing, and data analysis, while Computer Science covers scientific computing, software systems and engineering, computer engineering and systems as well as algorithms and theoretical computer science. Including an application field enables students to obtain in-depth knowledge and skills in an application area such as in the natural sciences, including medicine, biology, and physics, but also in the social sciences and humanities. Thus, this Master's program allows to cover all aspects from fundamental methods of Data Science and Computer Science to engineering-related aspects, and research and development in an application domain.

The Master's program allows a free choice of the course of study in order to facilitate an early entry into research-related as well as innovative practical subject areas. In particular, it allows the student to individualize the study program to a large extent, addressing particular needs and interest.

Current research topics and details about the Master's program in Data and Computer Science can be found on the website https://www.informatik.uni-heidelberg.de.

1.3 Subject-specific qualification objectives of the degree programme

The graduates of the Master's program in Data and Computer Science master in particular the competencies of Bachelor's graduates, in detail:

- They have knowledge in practical, theoretical, technical and applied Computer Science and the methods of mathematics and can apply these to solve concrete Computer Science problems.
- They can plan, carry out, document and present a Computer Science task self-reliantly.
- They can work on a problem from the field of Computer Science using scientific methods within a given period of time and develop and present proposed solutions.
- They master scientifically-based methods of programming and can apply them practically in projects. This includes the scientific methods of designing, implementing and debugging software.
- They know the concepts of designing and analyzing efficient algorithms and are able to use them when creating software.
- They know the basics of the use of operating systems and management of resources and are able to use this knowledge in the design, implementation, and optimization of computer systems.
- They know the problems and importance of reliability in modern computing systems and computer networks and are able to take this knowledge into account in the planning, implementation and control of such systems.

In addition, graduates of the Master's program in Data and Computer Science master the following professional qualifications beyond the learning outcomes of the Bachelor's program.

- They are able to independently plan, design and evaluate extensive computing systems under given technical and economic constraints and to manage associated software projects.
- They have in-depth knowledge in one or more special areas of Computer Science such as data analysis, requirements engineering, distributed systems, computer systems, and can apply this knowledge practically in the design and development of computing systems.
- They are able to decompose complex computing systems into abstract components (software and hardware) and determine and evaluate possibilities of realization according to given constraints, as well as to plan and implement this realization.
- They are able to independently familiarize themselves with future techniques of Computer Science, i.e. interdisciplinary areas, to apply them in projects, to communicate them professionally, and to develop them from a scientific point of view.

1.4 Generic qualification objectives of the degree programme

Graduates of the Master's program in Data and Computer Science should possess the following basic competencies of an interdisciplinary nature in the context of Computer Science.

- They possess problem-solving skills and are proficient in the application of knowledge in the field of Computer Science and additionally in a broader subject context or related disciplines. In addition, they are able to apply these skills in new, unfamiliar situations.
- They have the competence to work in a team as well as to take on more prominent responsibility in a team (team leadership).
- They are able to communicate their own conclusions based on the current state of research and application and to exchange ideas on a scientific level.
- They possess the competence to independently collect information, make judgments and independently acquire knowledge in the field of Computer Science as well as related disciplines. In particular, they are capable of procure and interpreting research literature and evaluating alternative solutions in the field of Computer Science as well as across disciplines.
- In addition, they are able to deal effectively with complex problems and situations, possess decision-making skills, and can independently carry out research- or application-oriented projects.
- They are able to communicate effectively in professional matters orally and in writing.

1.5 Particularities of the degree programme and module descriptions

1.5.1 Reason for modules with fewer than 5 credits

There are some modules in this program with less than 5 credit points. These modules are self-contained units of study in terms of content and cannot reasonably be combined with other modules.

1.5.2 Description of the teaching and learning forms

- **Lecture:** Presentation of the course content by the lecturer using appropriate media; interaction and questions are possible.
- **Exercise:** Exercises and smaller parts of the syllabus are explained; questions, interaction and discussion by and with the students to understand the syllabus and the example exercises.
- **Seminar:** Independent development of a scientific topic, preparation of a presentation, giving the presentation with subsequent questions and discussion of the participants about the presentation.
- **Practical:** Project work on the basis of a programming task, independent development of software including documentation, preparation of a project report and a lecture, giving a lecture on the presentation of the software.

1.5.3 Modalities for exams

At the beginning of each course, the details and, in particular, deviations from the modalities for exam listed below, will be announced by the lecturer oral and written.

Many modules have a uniform regulation for the awarding of the CP, so this regulation is described in detail here and then only referred to here in the module descriptions.

Rules for awarding the CP: In this module, the CP are awarded if the final exam is passed. The details of the final exam are described in the individual module descriptions. In this module there is a practice operation with the processing of exercises. In order to be admitted to the final exam, at least 50% of the points in the exercises must be achieved. This admission is valid for the current and the next two semesters (both examination periods each, see below), i.e. for modules offered annually, after admission, the final exam can be taken in this semester or one year later in both exam periods. After that, a renewed admission to the final examination in the practice operation must be aquired.

Examination scheme: This cell of the module description contains the number of attempts which are allowed to pass the module according to the examination regulations. A passed exam cannot be repeated.
1+1: after the first attempt there is only one repetition possibility .

Examination period: Two examination periods were set for the written examinations at the end of each semester. The first examination period lasts three weeks and consists of the last week of the lecture period and the first two weeks of the lecture-free period. The second examination period lasts three weeks and consists of the last three weeks of the lecture-free period. In exceptional cases, examinations can take place outside of these examination periods.

Examination dates: For modules that are offered once a year or less frequently, two examination dates are always offered after the module. In the case of written exams, these are within the examination periods mentioned above. In the case of oral exams, the dates are set by the lecturers. For modules that are offered every semester, there is only one examination date after the module. The students choose which of the offered examination dates they take.

If there are exceptions to the examination dates, especially if they are outside the examination periods mentioned above, the lecturer must announce them oral and written at the beginning of the course.

2 Model study plan and Mobility

2.1 Model study plan

1st year:		
	Elective area	44 CP
	Application Field	10 CP
	General competencies/Elective area	6 CP
sum		**60 CP**
2nd year:		
	Master Advanced Seminar	4 CP
	Master Advanced Practical	8 CP
	Elective area	6 CP
	Application Field	8 CP
	Master Thesis	30 CP
	Master Colloquium	4 CP
sum		**60 CP**
total:		**120 CP**

2.2 Mobility window

The mobility window for the Master's program in Data and Computer Science is usually located in the second and third semester, but a study visit to another university in Germany or abroad can also take place in the other semester. The Master's program contains only a few compulsory modules. Modules from the elective area or the application field tend to be easier to recognize through the elective options.

The planning for such a study visit should be started early, especially for a stay abroad, this organization phase can easily last a year.

Information on studying abroad can be found on the Erasmus program for computer science https://www.informatik.uni-heidelberg.de/erasmus.

3 Compulsory modules

The Master's program Data and Computer Science consists of the following compulsory modules (short summaries solely provided for improved understanding, for details please refer to the module handbook entries that follow afterwards):

- **Master Advanced Seminar:** the preparation and delivery of a scientific presentation of a seleted topic including a discussion
- **Master Advanced Practical:** practical work on a selected topic of advanced computer science, in particular recommended as preparation for the Master Thesis
- **Application Field:** obtaining in-depth knowledge and skills from one of the application areas listed in the examination regulations, thereby extending knowledge beyond the scope of computer science
- **Master Thesis:** the theoretical and practical work as well as thesis writing on an advanced scientific topic of computer science
- **Master Colloquium:** the presentation and defense of the results obtained during the work on the master thesis

They are described in the following.

Master Advanced Seminar

Code IMS	**Name** Master Advanced Seminar	
CP 4	**Duration** one semester	**Offered** each semester
Format 2 SWS seminar + 2 SWS tutorial	**Workload** 120h; thereof 30h presence study 90h preparation talk and report	**Availability** M.Sc. Data and Computer Science
Language German or English	**Lecturer(s)** depending on teaching offer	**Examination scheme** 1+1
Learning objectives	Students will deepen, practice and demonstrate - the ability to present advanced scientific literature and facts in a lecture in a factual and objective manner - knowledge of scientific writing techniques (including, in particular, literature research), and the ability to access advanced scientific literature - the advanced ability to discuss and give feedback on presentations - the ability to write a short and concise scientific paper on advanced scientific literature and issues - the advanced ability to provide feedback on scientific papers	
Learning content	- Improvement of scientific writing techniques and scientific feedback. - In-depth practice In the development and presentation of advanced scientific literature and topics - Selected advanced topics from computer science	
Requirements for participation	Recommended: knowledge in the topic of the seminar	
Requirements for the assignment of credits and final grade	The module is completed with a graded exam. This exam includes the preparation and delivery of a presentation of about 30-60 minutes (including discussion) as well as a written report of about 10 pages. More detailed regulations regarding the format of the paper and the presentation will be agreed upon at the beginning of the course. The exam must be passed in order to be awarded the LP. The final grade of the module is determined by the grade of the exam.	
Useful literature		

Master Advanced Practical

Code IMP	**Name** Master Advanced Practical	
CP 8	**Duration** one semester	**Offered** each semester
Format Practical 6 SWS	**Workload** 240h; thereof at least 25h presence 10h preparation presentation	**Availability** M.Sc. Data and Computer Science
Language German or English	**Lecturer(s)**	**Examination scheme** 1+1
Learning objectives	The students - acquire in-depth problem-solving competence for complex design and implementation tasks - are able to clearly present, demonstrate and apply problem analysis and description techniques - deepen programming knowledge in the respective programming language required for the project - are able to carry out the project with the help of a software development environment In addition, project-specific skills are deepened, especially working in a team (of up to three students): - Implementation and evaluation of projects - Planning and execution of project and team work The soft skills to be trained thus include in particular the ability to work in a team, refinement of presentation techniques, understanding scientific literature as well as independent work.	
Learning content	Domain knowledge dependent on lecturer; general learning content includes: - Deepening knowledge about the project's topic - Independent development of complex software and its documentation	
Requirements for participation		
Requirements for the assignment of credits and final grade	The module is completed with a graded exam. This exam includes the assessment of the project results (software, documentation), the project report (5-10 pages), and the presentation (approx. 30 minutes plus discussion). The note of this exam gives the note for this module. More details will be given by the lecturer.	

Useful literature	

Application Field

Code IAF	**Name** Application Field	
CP 18	**Duration**	**Offered**
Format Lecture, exercise, or practical	**Workload** 540h; division into attendance, practice and presence time in consultation with the lecturers	**Availability** M.Sc. Data and Computer Science
Language English or German	**Lecturer(s)** various, depending on application field	**Examination scheme**
Learning objectives	In-depth knowledge and skills in an application area	
Learning content	Selection of an application area according to the rules of the examination regulations. Determination of and participation in modules from the application area (CPs correspond to the specifications from the application area). It must be ensured that no modules from the application area are chosen that have already been taken in the Bachelor's program. (optional) Definition and implementation of an interdisciplinary project by a a lecturer from the application area and computer science. The project goal shall include a computer science achievement in the application area. Workload and thus CPs are determined by the lecturer. Contents shall be documented in a project report and a presentation.	
Requirements for participation	Recommended: same application field as in the bachelor studies	
Requirements for the assignment of credits and final grade	The examination credits can be obtained through non-informatics modules at bachelor's or master's level. Of the 18 CP, up to 10 CP can be earned through an interdisciplinary project. Examination credits in the application area and (optionally) for the interdisciplinary project (analogous to the module IMAP) are weighted according to the respective share of CP. Modules shall be graded, ungraded modules will only be admitted in justified exceptional cases.	
Useful literature		

Master Thesis

Code IMT	**Name** Master Thesis	
CP 30	**Duration** 6 month	**Offered** continuous
Format Supervised self-study	**Workload** 900 h processing of an individual topic (research and development work) and written elaboration	**Availability** M.Sc. Data and Computer Science
Language German or English	**Lecturer(s)** varying	**Examination scheme** 1+1
Learning objectives	- Use of the acquired technical knowledge and methods to independently solve a complex problem from computer science and its applications. - Ability to independently produce a scientific thesis	
Learning content	Independent scientific work on a demanding problem from the field of computer science and its applications	
Requirements for participation	45 CP (exam regulations - PO); elective modules, IMS and IMP recommended	
Requirements for the assignment of credits and final grade	Passing the graded master's thesis is required for the award of the LP. The Master's thesis includes regular consulting with advisor and the written elaboration.	
Useful literature	will be announced by the advisor	

Master Colloquium

Code IMC	**Name** Master Colloquium	
CP 4	**Duration**	**Offered** continuous
Format Colloquium	**Workload** 120h: Preparation presentation, guiding questions, and discussion; delivering presentation; defending discussion	**Availability** M.Sc. Data and Computer Science
Language German or English	**Lecturer(s)** depending on teaching offer	**Examination scheme** 1+1
Learning objectives	The students - acquire, practice and demonstrate the ability to present their own challenging work in a scientific presentation in an unbiased manner - gain skills and experience in defending advanced scientific topics - are able to position themselves clearly in their field, to communicate this, and, based on sound arguments, to defend the results of their own work in the context of the current state of the art in the context of a discussion	
Learning content	- Presentation of the content of the master thesis, especially the advantages and limitations as well as a comparison to the current state of the art. - Discussion, based on prepared guiding questions as well as open questions of different levels. Teachers as well as fellow students are allowed to participate in the discussion to cover thematically broadened views in terms of background and perspective - Content assessment of the paper is left to the examiner, with the focus of the colloquium assessment on the quality of the candidate's discussion and argumentation	
Requirements for participation	Completed master thesis (recommended)	
Requirements for the assignment of credits and final grade	The module is completed with a graded exam. This exam includes the evaluation of the presentation (approximately 30-60 minutes) and the student's ability to defend the results of his/her work in the face of questions and comments (approximately 15-45 minutes). Total time should not exceed 90 minutes. The exam must be passed in order to be awarded the LP. The final grade of the module is determined by the grade of the exam.	
Useful literature		

4 Elective modules

In the following, the elective modules of the Master's program Data and Computer Science are described. Specializations can (but do not have to) be chosen, in which case the following notes are to be observed. As described in the examination regulations, areas are to be covered when choosing modules. The assignment of the modules to the areas is described in the following. Subsequently, the descriptions of the specializations follow and, after this, the individual module descriptions.
Besides the modules from Computer Science (section 2.3), up to one Master Advanced Practical can be credited as elective module.

4.1 Module assignment to subject areas

In accordance with the specifications stated in the examination regulations, three of the following subject areas have to be covered with at least 6 CP each. The available areas as well as the modules assigned to these areas are listed in the following. For details about these modules, please refer to section 2.3 and following. Modules not listed in this subsection are not assigned to any specific area. The subject areas are as follows:

- Visual Computing (VC)
- Software Systems and Engineering (SE)
- Scientific Computing (SC)
- Algorithmic Data Analysis and Machine Learning (AM)
- Algorithmics and Theoretical Computer Science (AT)
- Computer Engineering (CE)

Module	VC	SE	SC	AM	AT	CE
3D Computer Vision (I3dCVi)	X					
Advanced Machine Learning (IAML)				X		
Algorithm Engineering (IAE)					X	
Articial Intelligence for Programming (IAIP)				X		
Complex Network Analysis (ICNA)					X	
Computational Geometry (ICGeo)	X					
Computerspiele (ICS)	X					
Convex Optimization			X			
Discrete Structures 2 (IDS2)					X	
Fundamentals of Machine Learning (IFML)				X		
Geometric Modeling and Animation (IGMA)	X					
Hardware Aware Scientic Computing (IHASC)			X			
IT-Projektmanagement (IPM)		X				
IT-Sicherheit (IITS)		X				
Inverse Probleme (IIP)			X			
Machine Learning (IML)				X		
Mining Massive Datasets (IMMD)				X		
Numerische Optimierung			X			
Optimization for Machine Learning (IOML)			X			
Praktische Geometrie (IPGeo)	X					
Scientic Visualization (ISV)	X					
Software Evolution (ISWEvol)		X				
Software Ökonomie (ISWÖk)		X				
Text Analytics (ITA)				X		
Volume Visualization (IVV)	X					
Wissensmanagement und Entscheidungen im Software Engineering (ISWKM)		X				
All basic & advanced modules of the MSc Computer Engineering (MScTI)						X

Table 4.1: Module assignment to subject areas

4.2 Specializations

Specializations represent a certain combination of elective modules, and ensure that the student has gained substantial knowledge in a certain area. If a certain combination is fulfilled, the specialization will be mentioned in the certificate of the Master degree. The following specializations exist, for more details regarding the combination of modules please refer to the following subsections.

- Specialization: Visual Computing
- Specialization: Information Systems Engineering
- Specialization: Scientific Computing
- Specialization: Algorithms and Theoretical Computer Science

4.2.1 Specialization: Visual Computing

This specialization qualifies for the development of concepts, algorithms, and software in visual computing. Visualization and computer vision cover techniques for data analysis. Computer games and animation address the generation of interactive and time-dependent content. The overall basis for the field is provided by computer graphics and geometry.

The specialization consists of seminars, practicals, as well as the bachelor and/or master thesis and elective modules. The choice of the elective modules must adhere to the following rules:

- 8 CT must be obtained from the lecture Computer Graphics (ICG), if not taken in the bachelor. Upon application, it can be granted to obtain these credit points from another lecture.
- At least 24 CT must be taken from the subject area Visual Computing (VC).

Module	**Semester**	**CP**
Master Advanced Seminar	1–3	4
Master Advanced Practical	1–3	8
Elective Modules	1–3	56
Master Thesis and Colloquium on the topic	4	34
Application Field	1–3	18
CP Sum		**120**

The point of contact for this specialization are **Susanne Krömker** and **Filip Sadlo**. It is recommend to obtain counseling before choosing this specialization, in particular with regard to the topic of the master thesis but also general module offering.

4.2.2 Specialization: Information Systems Engineering

This specialization comprises competencies to develop, operate and evolve complex information systems. It consists of seminars, practicals, master thesis and elective modules. The modules belong mainly to the groups data management and analysis (DMA, Michael Gertz), engineering mathematics and computing lab (EMCL, Vincent Heuveline), parallel and distributed systems (PVS, Artur Andrzejak) as well as software engineering (SWE, Barbara Paech). The core is a comprehensive practical (ISE project, winter semester, worth 16 CP), where the team develops for external customers an information system using software engineering methods for modern technologies. These comprise concepts and methods from data and text analysis, data management, information security, software quality and requirements engineering.

For this specialization the modules can be accredited according to this table:

Module	**Semester**	**CP**
Master Advanced Seminar	1 - 3	4
ISE project (covering two Master Practicals)	1 - 3	16
Elective modules	1 - 3	48
Master Thesis and Colloquium on the topic	4	34
Application Field	1 - 3	18
CP Sum		**120**

In addition the following rules apply:

- master advanced seminar and master thesis are taken from the above mentioned groups
- 3 different groups mentioned above from the focus area must be covered
- the choice of the elective modules satisfies:

  – at least 24 CT must be taken from the area Software Systems Engineering (SSE) or from the bachelor modules ISWQM or ISWRE

  – at least 6 CT must be taken from the area ADAML

  – at least 6 CT must be taken from the area ACTS

  – the elective modules can also be further seminars and practicals to the extent allowed by the examination regulations.

The point of contact for this specialization are **Michael Gertz (DMA)** and **Barbara Paech (SWE)**. It is recommend to obtain counseling before choosing this specialization, in particular with regard to the topic of the master thesis but also general module offering.

4.2.3 Specialization: Scientific Computing

This specialization can be chosen in the bachelor program, the master program as well as continuously in bachelor and master. The specialization consists of seminars, practicals, lectures as well as the bachelor and/or master thesis.

Modules in this specialization are

- Hardware Aware Scientic Computing (IHASC)
- Numerics (MD1)
- Introduction to optimization (MD3)
- Inverse Problems (IIP)
- Object-Oriented Programming for Scientic Computing (IOPSC)
- Optimization for Machine Learning (IOML)
- Convex Optimization
- Lectures from the build-up module "Numerics and Optimization" (MM25)
- Lectures from the specialization module "Numerics and Optimization" (MM35)

If this specialization has not been taken in the bachelor program, the module MD1 (numerics should be chosen.

Module	**Semester**	**CP**
Scholarly work	1	2
2 seminars in Scientific Computing	1 - 3	8
Numerics (MD1, if not taken in B.Sc.)	1 - 3	8
Other modules from Scientific Computing	1 - 3	24
Modules from at least two other specializations	1 - 3	26
Master Thesis and Colloquium on the topic	4	34
Application Field	1 - 3	18
CP Sum		**120**

The point of contact for this specialization is **Peter Bastian**. It is recommend to obtain counseling before choosing this specialization, in particular with regard to the topic of the master thesis but also general module offering.

4.2.4 Specialization: Algorithms and Theoretical Computer Science

This specialisation has two main aims. On the one hand, a deep understanding is gained for algorithms in real life applications with the key objective how these algorithms can be made as efficient as possible. On the other hand, investigations are made concerning theoretical questions of algorithms; this includes, mathematical theorems that give insights to considered structures that in turn yield faster algorithms in theory and practice.

To obtain this specialization, 50 credit points are required. Students can choose from the courses listed below.

Module	**Semester**	**CP**
Discrete Structures II	1 - 3	8
Algorithm Engineering	1 - 3	8
Master Advanced Seminar on Discrete Structures/Algorithm Engineering	1 - 4	4
Master Advanced Practical on Algorithm Engineering	1 - 3	8
Master Thesis and Colloquium on the topic	4	34

The point of contact for this specialization is **Felix Joos** and **Christian Schulz**. It is recommend to obtain counseling before choosing this specialization, in particular with regard to the topic of the master thesis but also general module offering.

4.3 Modules from computer science

The modules from computer science are described below in alphabetical order.

Advanced Machine Learning

Code IAML	**Name** Advanced Machine Learning	
CP 8	**Duration** one semester	**Offered** follows *Fundamentals of Machine Learning*
Format Lecture 4 SWS + Exercise course 2 SWS	**Workload** 240h, thereof 60h lecture 90h tutorials, homework, lecture wrap-up 90h graded final report	**Availability** cannot be combined with *Machine Learning* M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Ullrich Köthe	**Examination scheme** 1+1
Learning objectives	Students get to know advanced machine learning methods that define the state-of-the-art and major research directions in the field. Students understand when these methods are called for, what limitations of standard solutions they address, and how they are applied to real-world problems. In addition, students learn how to use Python-based machine learning software such as scikit-learn, theano and OpenGM.	
Learning content	The lecture, along with its sibling *Fundamentals of Machine Learning*, offers an extended version of the one-semester course *Machine Learning*: Multi-layered architectures (neural networks, deep learning); directed and undirected probabilistic graphical models (Gaussian processes, latent variable models, Markov random fields, structured learning); feature optimization (feature selection and learning, dictionary learning, kernel approximation, randomization); weak supervision (one-class learning, multiple instance learning, active learning, reinforcement learning)	
Requirements for participation	recommended are: lecture *Fundamentals of Machine Learning* or similar	
Requirements for the assignment of credits and final grade	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.	
Useful literature	David Barber: Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012 Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, 2006	

Algorithm Engineering

Code IAE	**Name** Algorithm Engineering	
CP 8	**Duration** one semester	**Offered** every summer semester
Format Lecture 4 SWS + Exercise course 2 SWS	**Workload** 240h; thereof 90h lectures and tutorials, 15h exam preparations, 135h lecture wrap-up and homework	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Christian Schulz	**Examination scheme** 1+1
Learning objectives	Students obtain a systematic understanding of algorithmic questions and solution approaches in the area of algorithm engineering. The students will be able to transfer the learned techniques onto similar problems and be able to interpret and understand current research topics in the area of algorithm engineering. Given a real-world problem, students are able to select appropriate algorithms to come up with and implement efficient solutions. In particular, students know realistic machine models and applications, algorithm design, implementation techniques, experimental methodology and can interpret of measurements.	
Learning content	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).	
Requirements for participation	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	
Requirements for the assignment of credits and final grade	The module is completed with a graded oral exam. The final grade of the module is determined by the grade of the exam. The requirements for the assignment of credits follows the regulations in section modalities for exams.	

Useful literature	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

Artificial Intelligence for Programming

Code IAIP	**Name** Artificial Intelligence for Programming	
CP 6	**Duration** one semester	**Offered** at least every 4th semester
Format Lecture 2 SWS + Exercise course 2 SWS	**Workload** 180 h; thereof 60 h lecture 15 h preparation for exam 105 h self-study and working on assignments (optionally in groups)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Artur Andrzejak	**Examination scheme** 1+1
Learning objectives	Expected learning outcomes are: - Knowledge of selected classical methods in arti-ficial intelligence, in particular knowledge repre-sentation, search methods, rule systems - Basic knowledge about probabilistic models and probabilistic programming - Knowledge of techniques for code representation and parsing - Knowledge of techniques for modeling of code via neural networks - Knowledge of basic and advanced methods for program synthesis - Familiarity with semantic parsing and code sum-marization - Familiarity with selected applications of AI for programming, e.g. code-to-code translation, code recommendations, and detection of bugs in code.	
Learning content	This module covers the following topics: - introduction to classical methods in artificial intelligence, in particular knowledge representation, search methods, rule systems - introduction to probabilistic models and probabilistic programming - fundamentals of code representation and parsing - modeling of code via neural networks and sequence models/transformers - basic and advanced methods for program synthesis - introduction to semantic parsing and code summarization - state-of-the-art applications of AI for programming, e.g. code-to-code translation, code recommendations, detection of vulnerabilities in code.	
Requirements for participation	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.	
Requirements for the assignment of credits and final grade	The module is completed with a graded oral or written exam. The final grade of the module is determined by the grade of the exam. The requirements for the assignment of credits follows the regulations in section modalities for exams.	

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

Complex Network Analysis

Code ICNA	**Name** Complex Network Analysis	
CP 8	**Duration** one semester	**Offered** every 2nd wintersemester
Format Lecture 4 SWS + Exercise course 2 SWS	**Workload** 240 h; thereof 90 h lecture 12 h preparation for exam 130 h self-study and working on assignments/projects (optionally in groups)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing B.Sc. Mathematik
Language English	**Lecturer(s)** Michael Gertz	**Examination scheme** 1+1
Learning objectives	Students - can describe basic measures and characteristics of complex networks - can implement and apply basic network analysis algorithms using programming environments such as R or Python - can describe different network models and can describe, compute, and analyze characteristic parameters of these models - know how to compute different complex network measures and how to interpret these measures - know different generative models for constructing complex networks, especially scale-free networks - know the fundamental methods for the detection of communities in networks and the analysis of their evolution over time - are familiar with basic concepts of network robustness - understand the principles behind the spread of phenomena in complex networks	
Learning content	- Graph theory and graph algorithms; basic network measures - Random networks and their characteristics (degree distribution, component sizes, clustering coefficient, network evolution), small world phenomena - Scale-free property of networks, power-laws, hubs, universality - Barabasi-Albert model, growth and preferential attachment, degree dynamics, diameter and clustering coefficient - Evolving networks, Bianconi-Barabasi model, fitness, Bose-Einstein condensation - Degree correlation, assortativity, degree correlations, structural cutoffs - Network robustness, percolation theory, attack tolerance, cascading failures - Communities, modularity, community detection and evolution - Spreading phenomena, epidemic modeling, contact networks, immunization, epidemic prediction	
Requirements for participation	recommended are: Algorithmen und Datenstrukturen (IAD), Knowledge Discovery in Databases (IKDD), Lineare Algebra I (MA4)	

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

Computational Geometry

Code ICGeo	**Name** Computational Geometry	
CP 8	**Duration** one semester	**Offered** irregular
Format Lecture 4 SWS + Exercise course 2 SWS	**Workload** 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)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Susanne Krömker	**Examination scheme** 1+1
Learning objectives	The students know the algorithms and data structures of geometric and topological data processing. They can understand and implement sweep algorithms for nearest neighbors, intersections of line segments and Voronoi diagrams, can construct alpha shapes and beta skeletons from pointclouds, know template-based and data-driven algorithms for the determination of isolines and isosurfaces, can work with discrete vector fields on simplicial complexes and know about persistence of topological invariants. They master the associated data structures for efficient storage and further processing and can calculate the complexity of the various algorithms.	
Learning content	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	
Requirements for participation	recommended is: Algorithmen und Datenstrukturen (IAD)	
Requirements for the assignment of credits and final grade	The module is completed with a graded oral exam. The final grade of the module is determined by the grade of the exam. The requirements for the assignment of credits follows the regulations in section modalities for exams.	
Useful literature	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	

Computerspiele

Code ICS	**Name** Computerspiele	
CP 8	**Duration** ein Semester	**Offered** jedes Sommersemester
Format Vorlesung 3 SWS + Übung 3 SWS	**Workload** 240 h; davon 75 h Präsenzstudium 15 h Prüfungsvorbereitung 150 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language Englisch	**Lecturer(s)** Jürgen Hesser	**Examination scheme** 1+1
Learning objectives	Die Studierenden lernen die Konzepte von der informatischen Sicht kennen, was Computerspiele ausmacht und welche Herausforderungen damit verbunden sind. Sie lernen, wie man gute und effiziente Architekturkonzepte dafür entwickelt sowie wie man typische Probleme aus Graphik, Kollisionserkennung, Animation/Physik, Pfadplanung/KI umsetzt. Zudem lernen sie in den Übungen, wie man konkrete Spiele entwickelt, so dass sie in der Lage sind, eigene Spieleengines zu realisieren.	
Learning content	Überblick über die Einteilung von Computerspielen Architektur von Game Engines Vorstellung von OGRE als einer open-source Game Engine Graphik und Computerspiele: ein Überblick Kollisionserkennungstechniken Animationstechniken und Physik bei Computerspielen mit Fokus auf der open source Bibliothek Bullet Pfadplanung und KI	
Requirements for participation	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
Requirements for the assignment of credits and final grade	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.	
Useful literature	Gregory et al: Game Engine Architecture Ericson: Real-Time Collision Detection Eberly: Game Physics Millington: Artificial Intelligence for Games	

Discrete Structures 2

Code IDS2	**Name** Discrete Structures 2	
CP 8	**Duration** one semester	**Offered** irregularly in the summer semester
Format Lecture 4 SWS + Exercise course 2 SWS	**Workload** 240 h; thereof 90 h lecture 20 h preparation for exam 130 h self-study and working on assignments/projects (optionally in groups)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Mathematik
Language English	**Lecturer(s)** Felix Joos	**Examination scheme** 1+1
Learning objectives	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	
Learning content	- Probabilistic Methods - Extremal graph theory - Expander graphs - Quasirandom graphs - Further advanced topics	
Requirements for participation	recommended is: Discrete Structures 1	
Requirements for the assignment of credits and final grade	The module is completed with a graded oral or written exam. The final grade of the module is determined by the grade of the exam. The requirements for the assignment of credits follows the regulations in section modalities for exams.	
Useful literature	- 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.	

Fundamentals of Machine Learning

Code IFML	**Name** Fundamentals of Machine Learning	
CP 8	**Duration** one semester	**Offered** in (irregular) alternation with *Machine Learning*
Format Lecture 4 SWS + Exercise course 2 SWS	**Workload** 240h, thereof 60h lecture 90h tutorials, homework, lecture wrap-up 90h graded final report	**Availability** cannot be combined with *Machine Learning* M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Ullrich Köthe	**Examination scheme** 1+1
Learning objectives	Students understand fundamental concepts of machine learning (features vs. response, unsupervised vs. supervised training, regression vs. classification etc.), get to know established learning methods and algorithms, are able to apply them to real-world problems, and can objectively assess the quality of the results. In addition, students learn how to use Python-based machine learning software such as scikit-learn.	
Learning content	The lecture, along with its sibling *Advanced Machine Learning*, offers an extended version of the one-semester course *Machine Learning*, with more room for regression methods, unsupervised learning and algorithmic details: Classification (nearest neighbor rules, linear and quadratic discriminant analysis, logistic regression, classical and randomized decision trees, support vector machines, ensemble methods); regression (linear and non-linear least squares, regularized and sparse regression, robust regression); unsupervised learning (hierarchical clustering, k-means algorithm, Gaussian mixture models and expectation maximization, principal component analysis, non-linear dimension reduction); evaluation (risk minimization, model selection, cross-validation)	
Requirements for participation	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
Requirements for the assignment of credits and final grade	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.	
Useful literature	Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning (2nd edition), Springer, 2009	

Geometric Modeling and Animation

Code IGMA	**Name** Geometric Modeling and Animation	
CP 8	**Duration** one semester	**Offered** every 3rd semester
Format Lecture 4 SWS + Exercise 2 SWS	**Workload** 240 h; thereof 90 h on-campus program 15 h exam preparation 135 h independent study and exercises (possibly in groups)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Filip Sadlo	**Examination scheme** 1+1
Learning objectives	The students know the mathematical foundations of geometric modeling know the mathematical and physical foundations of computer animation know the algorithms and implementation aspects are familiar with the basics of animated movies are able to apply existing tools for geometric modeling and animation	
Learning content	Introduction to curves Interpolating curves Bézier curves B-Splines Rational curves Introduction to surfaces Tensor product surfaces Transfinite surfaces and extrusion Subdivision Subdivision surfaces Animation and simulation Rigid body kinematics Particle systems Mass-spring models Cloth modeling Numerical methods for differential equations Collision detection and handling Fluid simulation and natural phenomena	
Requirements for participation	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	

Requirements for the assignment of credits and final grade	The module is completed with a graded oral or written exam. The final grade of the module is determined by the grade of the exam. The requirements for the assignment of credits follows the regulations in section modalities for exams.
Useful literature	- Curves and Surfaces for CAGD ? A Practical Guide, G. Farin, Morgan Kaufmann, 2002 - Computer Animation ? Algorithms and Techniques, R. Parent, Morgan Kaufmann, 2002 - 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, D. Eberly, Morgan Kaufmann, 2000 - Graphische Datenverarbeitung I, J. Encarnacao, W. Straßer, R. Klein, 4. Auflage, Oldenbourg 1996 - Advanced Animation and Rendering Techniques, A. Watt, M. Watt, Addison-Wesley, 1992 - Grundlagen der geometrischen Datenverarbeitung, J. Hoschek, D. Lasser, Teubner 1992 - Numerical Recipes ? The Art of Scientific Computing, W.H. Press, P. Flannery, S.A. Teukolsky, W.T. Vetterling, Cambridge University Press, 1986

Hardware Aware Scientific Computing

Code IHASC	**Name** Hardware Aware Scientific Computing	
CP 8	**Duration** one semester	**Offered** irregular
Format Lecture 4 SWS + Exercise Course 2 SWS	**Workload** 240h;thereof 90h lecture 15h preparation for exam 135h self-study and working on assignments/projects (optionally in groups)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Peter Bastian	**Examination scheme** 1+1
Learning objectives	Students are familiar with different forms of parallelism in modern computer architectures. The can exploit this parallelism selecting an appropriate programming model. They are familiar with modelling of parallelism and know fundamental parallel algorithms from scientific computing.	
Learning content	Parallel Computer Architecture - Pipelining and super-scalar processors, SIMD vectorisation - Caches - Multicore architectures - GPUs - Communication networks Programming Models - Shared memory programming with OpenMP and C++ threads - OpenCL or Cuda - Task-based programming - Message-passing, MPI Parallel Algorithms - Speedup & scalability - Roofline model - Linear Algebra: Matrix-Vector, Matrix multiplication, solving dense systems, solving sparse systems - Iterative Solution of Linear Systems - High-Performance Libraries - Differential equations - Particle Methods	
Requirements for participation	basic knowledge in computer architecture and numerical methods; good programming skills in C++	

Requirements for the assignment of credits and final grade	The module is completed with a graded exam. The note of this exam gives the note for this module. Details for this exam as well as the requirements for the assignment of credits will be given by the lecturer an the beginning of this course.
Useful literature	Frédéric Magoules, François-Xavier Roux, Guillaume Houzeaux: Parallel Scientific Computing, Wiley, 2016, doi: 10.1002/9781118761687

IT-Projektmanagement

Code IPM	**Name** IT-Projektmanagement	
CP 3	**Duration** ein Semester	**Offered** jedes 2. Wintersemester
Format Vorlesung+Übung 2 SWS	**Workload** Vorlesung + Übung 90 h insgesamt, davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen) für eine Anrechnung als EPG zusätzlich: 90 h Forschungsprojekt, davon 15 h Einarbeitung und Literaturrecherche 15 h Vorbereitung einer Studie, z.B. Interview-Training 20 h Durchführung einer Studie 20 h Auswertung 20 h Abschlusspräsentation + Berichterstellung	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Language Englisch	**Lecturer(s)** Andrea Herrmann	**Examination scheme** 1+1
Learning objectives	Die Teilnehmer/innen können ein Projekt planen und überwachen, verstehen, wie Projekte in Organisationen eingebettet sind und haben Grundkenntnisse in vertraglichen Themen. Die Teilnehmer/innen des EPG können ein Forschungsprojekt selbständig durchführen und kennen Forschungsergebnisse aus der Gender-Forschung.	
Learning content	Projektplanung, Projektorganisation Kostenschätzung Angebot/ Vertrag, Verhandeln Vorgehensmodelle Risikomanagement Controlling IT-Vertragsrecht Änderungsmanagement Zeitmanagement Projektabschluss Verteilte Softwareentwicklung	
Requirements for participation	keine	

Requirements for the assignment of credits and final grade	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.
Useful literature	PMI (Project Management Institute): A Guide to the Project Management Body of Knowledge (PM BOK ® Guide), 4. Ausgabe 2008

IT-Sicherheit

Code IITS	**Name** IT-Sicherheit	
CP 8	**Duration** ein Semester	**Offered** unregelmäßig
Format Vorlesung 4 SWS + Übung 2 SWS	**Workload** 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Language Deutsch oder Englisch	**Lecturer(s)** Vincent Heuveline	**Examination scheme** 1+1
Learning objectives	Studierende * erwerben umfangreiches Wissen über die Funktionsweise und Verwundbarkeiten vernetzter Computersysteme, und können somit Konzepte zur IT-Netzsicherheit bewerten und entwerfen * erlangen grundlegende Kenntnisse über die Sicherung großer Netzwerke und der Kommunikationsinfrastruktur (Routing, Namensauflösung, Internet-Firewalls, Intrusion Detection Systeme) * erwerben Kenntnisse im Bereich Kryptographie: Theorie der Kryptographie und praktische Umsetzung typischer kryptographischer Verfahren im Zusammenhang mit kryptographischen Prüfwerten, symmetrischen und asymmetrischen Chiffrierverfahren * erwerben umfassende Kompetenzen zur Detektion von Cyberangriffen; grundlegende Kompetenzen im Feld der IT-/Cyber-Forensik * erwerben praktische Erfahrungen bei der Verwendung von dedizierter Software zur Detektion von Angriffsszenarien im Datennetz * erwerben grundlegender Kenntnisse zum BSI IT-Grundschutzkonzept * erwerben grundlegender Kenntnisse im Penetrationtesting und dazugehöriger Werkzeuge Langfristiges Ausbildungsziel: Einsatz-/Beschäftigungsfähigkeit in der Breite des Arbeitsfeldes IT-Sicherheit;	

Learning content	Der IT-Sicherheit kommt bei der allgegenwärtigen Digitalisierung eine Schlüsselrolle zu. Diese Vorlesung vermittelt methodische Ansätze zur Modellierung und Bewertung von Angriffsszenarien, auf Basis welcher wirksame technische Gegenmaßnahmen umgesetzt werden können. Insbesondere werden folgende Schwerpunkte adressiert: - Sicherheitsmodelle und Bewertungskriterien - Kryptographische Prüfwerte: Modifikationserkennungs- und Nachrichtenauthentisierungswerte - Symmetrische und asymmetrische kryptographische Verfahren - Kryptographische Protokolle - Zugangs/Zutritts/Zugriffskontrolle - Schutz von Kommunikationsinfrastrukturen - Digitale Identität - Penetrationtesting / Schwachstellenanalyse - BSI IT-Grundschutzbausteine Mit Hilfe von virtuellen Maschinen in einem geschützten Bereich werden klassische Angriffs- und Schutzszenarien praktisch untersucht. Hier werden sog. Experimentierblätter samt Daten zur Verfügung gestellt.
Requirements for participation	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Betriebssysteme und Netzwerke (IBN), Einführung in die Technische Informatik (ITE), Algorithmen und Datenstrukturen (IAD)
Requirements for the assignment of credits and final grade	Das Modul wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.
Useful literature	C. Eckert, IT-Sicherheit: Konzepte, Verfahren, Protokolle De Gruyter Studium. Oldenbourg: de Gruyter, 2014. T.W. Harich, IT-Sicherheitsmanagement: Arbeitsplatz IT Security Manager. MITP, 2012. J.P.Müller, Security for Web Developers, O?Reilly, 2018

Inverse Problems

Code IIP	**Name** Inverse Problems	
CP 8	**Duration** ein Semester	**Offered** jedes Sommersemester
Format Vorlesung 2 SWS, Übung 2 SWS und Hausarbeiten	**Workload** 240 h; davon 60 h Präsenzstudium 15 h Prüfungsvorbereitung 165 h Selbststudium und Aufgabenbearbeitung, Hausarbeiten	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Language Englisch	**Lecturer(s)** Jürgen Hesser	**Examination scheme** 1+1
Learning objectives	Die Studierenden verstehen, was inverse Probleme sind und warum sie schwer zu lösen sind. Sie lernen die Prinzipien kennen, wie man sowohl deterministische Probleme, als auch stochastische Probleme löst und dabei auch die Regularisierungsparameter geeignet wählt. Schließlich erfahren sie die neuesten Entwicklungen im Bereich compressed sensing. Alle Prinzipien werden an zwei ausgewählten Gebieten, der Tomographie und des Deblurrings dargestellt. Sie erhalten damit die Kompetenz komplexe Probleme zu lösen, die mit klassischen Techniken nicht stabil lösbar sind und sind damit in der Lage auch komplexe experimentelle Messungen adäquat auswerten zu können.	
Learning content	Deterministische inverse Probleme Stochastische inverse Probleme Wahl der Regularisierungsparameter Compressed sensing Tomographie Deblurring	
Requirements for participation	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Numerische Mathematik	
Requirements for the assignment of credits and final grade	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.	
Useful literature	M. Bertero, P. Boccacci: Introduction to Inverse Problems in Imaging, IoP, 2002 web-Page and book: http://www.slaney.org/pct/pct-toc.html	

Machine Learning

Code IML	**Name** Machine Learning	
CP 8	**Duration** one semester	**Offered** in (irregular) alternation with *Fundamentals of Machine Learning* + *Advanced Machine Learning*
Format Lecture 4 SWS + Exercise course 2 SWS	**Workload** Arbeitsaufwand: 240h, thereof 60h lecture 90h tutorials, homework, lecture wrap-up 90h graded final report	**Availability** 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
Language English	**Lecturer(s)** Ullrich Köthe	**Examination scheme** 1+1
Learning objectives	Students understand a broad range of machine learning concepts, get to know established and advanced learning methods and algorithms, are able to apply them to real-world problems, and can objectively assess the quality of the results. In addition, students learn how to use Python-based machine learning software such as scikit-learn.	
Learning content	This lecture is a compact version of the two-semester course *Fundamentals of Machine Learning* + *Advanced Machine Learning*: Classification (linear and quadratic discriminant analysis, neural networks, linear and kernelized support vector machines, decision trees and random forests), least squares and regularized regression, Gaussian processes, unsupervised learning (density estimation, cluster analysis, Gaussian mixture models and expectation maximization, principal component analysis, bilinear decompositions), directed probabilistic graphical models, optimization for machine learning, structured learning	
Requirements for participation	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
Requirements for the assignment of credits and final grade	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.	

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

Mining Massive Datasets

Code IMMD	**Name** Mining Massive Datasets	
CP 6	**Duration** one semester	**Offered** at least every 4th semester
Format Lecture 2 SWS + Exercise course 2 SWS	**Workload** 180 h; thereof 60 h lecture 15 h preparation for exam 105 h self-study and working on assignments (optionally in groups)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Artur Andrzejak	**Examination scheme** 1+1
Learning objectives	* Knowledge of selected approaches and programming paradigms of parallel data processing * Knowledge how to use tools for parallel data processing (among others Apache Hadoop and Spark) * Familiarity with application domains of big data analysis * Knowledge of methods of parallel pre-processing of data * Knowledge of methods like classification, regression, clustering and their parallel implementations * Knowledge of scaling of parallel algorithms	
Learning content	This module covers the following topics: * programming paradigms for parallel-distributed data processing, especially Map-Reduce and Spark programming models * usage of tools like Apache Spark, Hadoop, Pig, Hive, and possibly other frameworks for parallel-distributed data processing * application cases in parallel data analysis, for example clustering, recommendation, search for similar objects, mining of data streams * techniques for parallel pre-processing of data * fundamentals of analysis techniques such as classi&#64257;cation, regression, clustering and evaluation of the results * parallel algorithms for data analysis and their implementations * theory and practice of scalability and tuning of frameworks	
Requirements for participation	recommended are Knowledge of Java/Python and in elementary probability theory / statistics; module IBD can be taken as a complement / extension.	
Requirements for the assignment of credits and final grade	The module is completed with a graded exam. This note of this exam gives the note for this module. Details for this exam as well as the requirements for the assignment of credits will be given by the lecturer an the beginning of this course.	

Useful literature	* Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, Mining of Massive Datasets, Cambridge University Press, Version 2.1 von 2014 (http://www.mmds.org/) * Trevor Hastie, Robert Tibshirani, Jerome Fried-man, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009 (http://statweb.stanford.edu/~tibs/ElemStatLearn/) * Ron Bekkerman, Misha Bilenko, John Langford, Scaling Up Machine Learning, Cambridge University Press, 2012 * Jiawei Han, Micheline Kamber, Jian Pei, Data Mining: Concepts and Techniques, Morgan Kaufmann, (third edition), 2012 * Books from O'Reilly Data Science Starter Kit, 2014 (http://shop.oreilly.com/category/get/data-science-kit.do)

Optimization for Machine Learning

Code IOML	**Name** Optimization for Machine Learning	
CP 8	**Duration** one semester	**Offered** every winter semester
Format Lecture 4 SWS + Exercise course 2 SWS	**Workload** 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)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Mathematik M.Sc. Scientific Computing
Language English	**Lecturer(s)** Bogdan Savchynskyy	**Examination scheme** 1+1
Learning objectives	The students - can analyze optimization methods for machine learning problems and estimate the area of their potential application - can competently apply existing algorithms and program packages for inference and learning with graphical models and neural networks - know typical optimization techniques for inference and learning with graphical models and neural networks - understand the basics of convex analysis, convex optimization, convex duality theory, (integer) linear programs and their geometry	
Learning content	The course presents various existing optimization techniques for such important machine learning tasks, as inference and learning for graphical models and neural networks. In particular, it addresses such topics as combinatorial algorithms, integer linear programs, scalable convex and non-convex optimization and convex duality theory. Graphical models and neural networks play a role of working examples along the course. The content of the course includes: - Convex analysis and optimization: convex sets and functions, polyhedra, (integer) linear programs, basic first-order convex optimization methods and their stochastic variants, LP and Lagrange relaxations - Graphical Models: dynamic programming, sub-gradient and block-coordinate ascent inference methods, min-cut/max-flow based inference, structured risk minimization for graphical models - neural networks: architectures, backpropagation algorithm, stochastic gradient descent and its variants for training neural networks.	
Requirements for participation	recommended are: linear algebra, analysis and any universal programming language (e.g. C/C++/Pascal/python)	

Requirements for the assignment of credits and final grade	The module is completed with a graded oral exam. The final grade of the module is determined by the grade of the exam. The requirements for the assignment of credits follows the regulations in section modalities for exams.
Useful literature	will beannounced by the lecturer at the beginning of the course

Praktische Geometrie

Code IPGeo	**Name** Praktische Geometrie	
CP 4	**Duration** ein Semester	**Offered** unregelmäßig
Format Vorlesung 2 SWS, Übung 1 SWS	**Workload** 120 h; davon 45 h Präsenzstudium 60 h Aufgabenbearbeitung 15 h Prüfungsvorbereitung	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language Deutsch	**Lecturer(s)** Susanne Krömker	**Examination scheme** 1+1
Learning objectives	Verständnis grundlegender geometrischer Konzepte zur Datenanalyse sowie effektive Punktsuche und Weiterverarbeitung von Messdaten Souveräner Umgang mit Projektionen und Beschreibungen jenseits der dreidimensionalen Erfahrungswelt Berechnung geometrischer Invarianten, Distanzen, Krümmungen aus Messdaten, rekonstruierten und generierten Flächen	
Learning content	Grundlegende Gebiete der Geometrie mit Relevanz in Computergraphik, Bildverarbeitung, Mustererkennung, Computer Vision und Geometrischem Modellieren (i) Analytische Geometrie: Operationen auf Vektorräumen mit geeigneten Koordinaten und Abbildungen (Affinitäten, Kollinearitäten), geometrische Ausgleichsprobleme aus fehlerbehafteten Messdaten (ii) Projektive Geometrie: Zentralprojektion und inverse Rekonstruktion von 3D-Objekten aus ebenen Bildern (Computer Vision, Geodäsie), Unterschiede zwischen B-Spline-Kurven und -Flächen und der Klasse der NURBS, Freiformgeometrien in CAD-Systemen (iii) Differentialgeometrie: Parameterdarstellungen in der geometrischen Datenverarbeitung, implizite Darstellungen (level sets), Abschätzung von Invarianten aus diskreten Daten (Triangulierungen, Punktwolken)	
Requirements for participation	empfohlen sind: Einführung in die Praktische Informatik (IPI), Mathematik für Informatiker (IMI1 und 2) oder Lineare Algebra (MA4)	
Requirements for the assignment of credits and final grade	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.	
Useful literature	Geometrie für Informatiker, Skript TU Wien 2004, Helmut Pottmann Aktuelle Fachveröffentlichungen	

Projektseminar Biomedizinische Bildanalyse

Code IPBB	**Name** Projektseminar Biomedizinische Bildanalyse	
CP 6	**Duration** ein Semester	**Offered** jedes Sommersemester
Format 2 Teile Seminar und Projekt, 4 SWS	**Workload** 180 h (je zur Hälfte Seminar und Projekt) 60 h Präsenzstudium 120 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	**Availability** B.Sc. Angewandte Informatik M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Language Deutsch	**Lecturer(s)** Karl Rohr	**Examination scheme**
Learning objectives	Die Studierenden erlangen vertiefte Kenntnisse und Fähigkeiten im Gebiet Biomedizinische Bildanalyse lernen fortgeschrittene Methoden und Algorithmen zur automatischen Analyse biomedizinischer Bilder lernen wie man Algorithmen und Software für automatische Bildanalyse entwickelt erweitern ihre Fähigkeiten Projektergebnisse mündlich zu präsentieren und schriftlich zu dokumentieren erweitern ihre Fähigkeiten zur Teamarbeit und zur Strukturierung von Projekten	
Learning content	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).	
Requirements for participation	empfohlen sind: Grundkenntnisse in Bildverarbeitung (Computer Vision, Image Analysis), Programmierkenntnisse, Kenntnisse in Software Engineering	
Requirements for the assignment of credits and final grade	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.	

Useful literature	Bekanntgabe in der Lehrveranstaltung

Scientific Visualization

Code ISV	**Name** Scientific Visualization	
CP 8	**Duration** one semester	**Offered** every 3rd semester
Format Lecture 4 SWS + Exercise 2 SWS	**Workload** 240 h; thereof 90 h on-campus program 15 h exam preparation 135 h independent study and exercises (possibly in groups)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Filip Sadlo	**Examination scheme** 1+1
Learning objectives	The students understand fundamental and advanced concepts of scientific visualization. They understand the mathematical fundamentals, data structures, and implementation aspects. They get to know schemes for interpolation and integration, mapping for scalar, vector, and tensor fields, and derived approaches. The students understand approaches for direct and indirect volume rendering, feature extraction, and topology-based analysis. The students are able to apply these concepts to real-world problems using existing software packages, and develop small programs using visualization libraries.	
Learning content	- Introduction - Visualization Process - Data Sources and Representation - Interpolation and Filtering - Approaches for Visual Mapping - Scalar Field Visualization: Advanced Techniques for Contour Extraction, Classification, Texture-Based Volume Rendering, Volumetric Illumination, Advanced Techniques for Volume Visualization, Pre-Integration, Cell Projection, Feature Extraction - Vector Field Visualization: Vector Calculus, Particle Tracing on Grids, Vector Field Topology, Vortex Visualization, Feature Extraction, Feature Tracking - Tensor Field Visualization: Glyphs, Hue-Balls and Lit-Tensors, Line-Based Visualization, Tensor Field Topology, Feature Extraction	
Requirements for participation	strongly recommended is: Computer Graphics (ICG) recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
Requirements for the assignment of credits and final grade	The module is completed with a graded oral or written exam. The final grade of the module is determined by the grade of the exam. The requirements for the assignment of credits follows the regulations in section modalities for exams.	

Useful literature	C.D. Hansen, C.R. Johnson, The Visualization Handbook, 2005.

Scientific Working

Code ISCW	**Name** Scientific Working	
CP 2	**Duration** one semester	**Offered** each winter
Format Lecture 2 SWS	**Workload** 60 h; thereof 30 h presence study 30 h own studies and practical exercises	**Availability** M.Sc. Data and Computer Science
Language German or English	**Lecturer(s)** varying	**Examination scheme** 1+1
Learning objectives	The students - know the most important literature sources in computer science; - know which tools and techniques exist for managing literature and how to use them; - are able to critically read and evaluate scientific texts (e.g. from conference proceedings or journals) and presentations and summarize them compactly; - know the relevant techniques for presenting a scientific paper; - know the possibilities of scientific publishing and the organization of scientific meetings; - know different research methods; - are familiar with current research in computer science - have an overview of the ways of financing research work; - know the requirements for the structure of applications for research funding.	
Learning content	- Literature research and management - Scientific presentation, writing, publishing and reviewing - Research funding via third-party funds - Research methods and current research projects - Scientific work after graduation	
Requirements for participation	none	
Requirements for the assignment of credits and final grade	The module is completed with a graded exam. The final note of this module is determined by the note of this exam. Details for this exam as well as the requirements for the assignment of credits will be given by the lecturer an the beginning of this course.	
Useful literature		

Software Evolution

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

Requirements for the assignment of credits and final grade	Das Modul wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.
Useful literature	Arnold, R. (Hrsg.): Software Reengineering. IEEE Computer Socienty Press, Los Alamitos 1993 Fowler, M.: Refactoring ? Improving the Design of Existing Code. Addison-Wesley, Reading, Massachusetts, 1999 von Hahn, E.: Werterhaltung von Software. DUV, Wiesbaden 2005 Müller, B.: Reengineering. Eine Einführung. Teubner, Stuttgart 1997 Sneed, H.M.; Hasitschka, M.; Teichmann, M.-T.: Software-Produktmanagement. Wartung und Weiterentwicklung bestehender Anwendungs-systeme. dpunkt, Heidelberg 2005 Smith, D.D.: Designing Maintainable Software. Springer, Heidelberg 1999

Software Ökonomie

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

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

Text Analytics

Code ITA	**Name** Text Analytics	
CP 8	**Duration** one semester	**Offered** every 2nd winter semester
Format Lecture 4 SWS + Exercise course 2 SWS	**Workload** 240 h; thereof 90 h lecture 15 h preparation for exam 135 h self-study and working on assignments/projects (optionally in groups)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Michael Gertz	**Examination scheme** 1+1
Learning objectives	Students - can implement and apply different text analytics methods using open source NLP and machine learning frameworks - can describe different document and text representation models and can compute and analyze characteristic parameters of these models - know how to determine, apply, and interpret use-case specific document similarity measures and underlying ranking concepts - know the concepts and techniques underlying different text classification and clustering approaches - know different models for phrase extraction and text summarization and are able to apply respective models and concepts using NLP and machine learning frameworks - know the fundamental methods for the extraction of document outlines at different levels of granularity - are familiar with basic concepts of topic models and their application in different text analytics tasks - understand the principles of evaluating results of text analytics tasks - know the theoretical background of machine learning methods at sufficient depths to be able to choose parameters and adapt an algorithm to a given text analytics problem - are aware of ethical issues arising from applying text analytics in different domains	

Learning content	- Text analytics in the context of Data Science - Open source text analytics, NLP, and machine learning frameworks - Fundamentals of NLP pipeline components - Document and text representation models - Document and text similarity metrics - Approaches, techniques and corpora for benchmarking text analytics tasks - Traditional and recent text classification and clustering approaches - Information extraction and topic detection approaches - Fundamentals of keyword and phrase extraction - Text summarization techniques - Generating document and text outlines - Ethical and legal aspects of text analytics methods - Text Analytics project management
Requirements for participation	Recommended are: solid knowledge of basic calculus, statistics, and linear algebra; good Python programming skills
Requirements for the assignment of credits and final grade	The module is completed with a graded exam. The note of this exam gives the note for this module. Details for this exam as well as the requirements for the assignment of credits will be given by the lecturer an the beginning of this course.
Useful literature	The following textbooks and texts are useful but not required. - Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft) - Yoav Goldberg. A Primer on Neural Network Models for Natural Language Processing (2015) - Christopher D. Manning and Hinrich Schütze: Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: May 1999. Furthermore, during the course of this lecture, several papers covering topics discussed in class will be provided.

Volume Visualization

Code IVV	**Name** Volume Visualization	
CP 8	**Duration** one semester	**Offered** every summer semester
Format Lecture 4 SWS + Exercise course 3 SWS	**Workload** 240 h	**Availability** M.Sc. Angewandte Informatik, M.Sc. Data and Computer Science,
Language English	**Lecturer(s)** Jürgen Hesser	**Examination scheme** 1+1
Learning objectives	The students 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. They further understand the physical principles of volume rendering, the different strategies of their realization with advantages and disadvantages ? they should critically assess different techniques ? and their parallelization.	
Learning content	Introduction of the visualization of scientific data of natural sciences and bio-sciences Discrete and continuous representation of data and methods of interpolation Methods of conversion between surface- and volumerepresentations and their efficient realizations Theory of volume rendering and their different realizations Accelerating and parallelization of volume rendering Programming technique: GPU-programming	
Requirements for participation	recommended are: Introduction into computer science I (IPI), programming course (IPK), algorithms & data structures (IAD);	
Requirements for the assignment of credits and final grade	The module is completed with a graded written exam. The final grade of the module is determined by the grade of the exam. The requirements for the assignment of credits follows the regulations in section modalities for exams.	
Useful literature	Engel et al.: Real-Time Volume Graphics www.real-time-volume-graphics.org, Schroeder et al.: VTK Textbook http://www.kitware.com/products/books/vtkbook.html	

Wissensmanagement und Entscheidungen im Software Engineering

Code ISWKM	**Name** Wissensmanagement und Entscheidungen im Software Engineering	
CP 3	**Duration** ein Semester	**Offered** jedes 2. Wintersemester
Format Vorlesung+Übung 2 SWS	**Workload** 90 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Language Deutsch	**Lecturer(s)** Andrea Herrmann	**Examination scheme** 1+1
Learning objectives	Die Teilnehmer/innen kennen vertiefende Software Engineering Methoden, die Entscheidungen unterstützen bei Anforderungspriorisierung, Entwurf, Managemententscheidungen und Risikomanagement. Sie wissen, wie man im Arbeitsalltag Wissen verwaltet und haben eine Einführung in die Entscheidungstheorie erhalten.	
Learning content	Wissensmanagement Ontologien Rationale Re-engineering learning organization Entscheidungen Management-Entscheidungen, Business Case Risikomanagement Anforderungspriorisierung Entscheidungen im Entwurf: ATAM, SAAM, CBAM Entscheidungstheorie Entscheiden unter Ungewissheit Mathematical Economics Entscheidung mit mehreren Parteien: Harvard-Konzept, Verhandlungen Spieltheorie Fehlentscheidungen/ Decision Traps/ Biases	
Requirements for participation	empfohlen sind: Vorlesung und Übung Einführung in Software Engineering (ISW) oder vergleichbare Vorkenntnisse	
Requirements for the assignment of credits and final grade	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.	

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

4.4 Modules from BSc/MSc Mathematics

4.4.1 Bachelor of Mathematics

The following modules from the Bachelor of Mathematics with 100% subject content can be credited:

- Probability Theory (MC4)
- Numerics (MD1)
- Statistics (MD2)
- Introduction to Optimization (MD3)
- Computational Statistics (MD6)
- Mathematical Logic (ME3)

4.4.2 Master of Mathematics

From the Master of Mathematics, the following courses from the following modules can be credited.

Basic Module Numerics and Optimization (MM15)

- Finite Elements
- Nonlinear Optimization
- Numerical Optimization with Differential Equations I
- Uncertainty Quantification 1

Basic Module Statistics and Probability

- Probability Theory II
- Statistics II

Advanced Module Numerics and Optimization (MM25)

- Mixed Finite Elements
- Parallel solvers for finite elements
- Numerical Optimization with Differential Equations II
- Uncertainty Quantification 2

Specialization Module Numerics and Optimization (MM35)

- Computational Fluid Dynamics
- Fundamentals of Computational Environmental Physics
- Mathematical Methods of Image and Pattern Analysis II Convex Optimization
- Numerical Methods for Bayesian Inverse Problems
- Numerical Methods for Maxwell's Equations

Specialization Module Statistics and Probability (MM36)

- Statistical Analysis of Machine Learning Algorithms

From the supplementary modules

- Computability and Complexity I
- Computability and Complexity II

4.5 Modules from the MSc Computer Engineering

All subject-related modules from the MSc Computer Engineering can also be credited in the MSc Data and Computer Science according to the content requirements. The modules offered can be found in the current module handbook of the Master of Computer Engineering.