Module Handbook

Master Course of Studies
”Data and Computer Science”

Universität Heidelberg
Fakultät für Mathematik und Informatik

Version as of 15.10.2025, corresponding to examination regulations of 22.07.2010 with changes dated 03.02.2013, 29.09.2021 and 05.10.2022

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: 50

Fee: According to general regulations of Heidelberg University

Contents

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

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 program

The master's program 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 applied field enables students to obtain in-depth knowledge and skills in an applied 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 interests.

Current research topics and details about the master's program 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 program

The graduates of the master's program Data and Computer Science 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 Data and Computer Science 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 program

Graduates of the master's program 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 responsibilities 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 judgements and independently acquire knowledge in the field of computer science as well as related disciplines. In particular, they are capable of procuring 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 program and module descriptions

1.5.1 Reason for modules with less than 5 credit points

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 presentation of the project.

1.5.3 Modalities for examinations

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

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

Rules for awarding CP. In this module, CP are awarded if the final examination is passed. The details of the final examination are described in the individual module descriptions. Exercises are processed in a group with a tutor. In order to be admitted to the final examination, 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 examination 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 exercise group must be acquired.

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. Once an exam is passed, it cannot be repeated in order to improve the grade. **1+1:** after the first attempt, there is only one repetition possible.

Examination period. There are two examination periods for written examinations at the end of each semester. The first examination period consists of the last week of the lecture period and the first two weeks of the lecture-free period. The second examination period consists of the last two weeks of the lecture-free period and the first week of the following lecture period. In exceptional cases, examinations can take place out of these examination periods.

Examination dates. For modules offered once a year or less frequently, two examination dates are offered after the end of the module. Written exams are offered within the examination periods mentioned above. Oral exams are set by the lecturers. For modules offered every semester, there is only one examination date after the end of the module. The students choose themselves 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 orally 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 / Elective Area	10 CP
	General competencies / Elective area	6 CP
sum		**60 CP**
2nd year:		
	Master's Advanced Seminar	4 CP
	Master's Advanced Practical	8 CP
	Elective Area	6 CP
	Application Field / Elective Area	8 CP
	Master's Thesis	30 CP
	Master's Colloquium	4 CP
sum		**60 CP**
total:		**120 CP**

2.2 Mobility window

The mobility window for the master's program Data and Computer Science is usually located in the second and third semester, nevertheless a study visit to another university in Germany or abroad can also take place in another semester. The planning of such a study visit should be started early. Especially for a stay abroad, the organization phase can easily take 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's Advanced Seminar:** the preparation and delivery of a scientific presentation of a seleted topic including a discussion.
- **Master's Advanced Practical:** practical work on a selected topic of advanced computer science, in particular recommended as preparation of the master's 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. Optionally, it is possible to conduct the application field using modules from the master's program Data and Computer Science.
- **Master's Thesis:** theoretical and practical work as well as thesis writing on an advanced scientific topic of computer science.
- **Master's Colloquium:** presentation and defense of the results obtained during the work on the master's thesis.

They are described in the following.

Master’s Advanced Seminar

LV-Nr. 110024XXXX	**Name** Master’s Advanced Seminar	**Abbreviation** IMS
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 examination. This examination 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 examination must be passed in order to be awarded the CP. The final grade of the module is determined by the grade of the examination.	
Useful literature		

Master's Advanced Practical

LV-Nr. 1100253XXX	**Name** Master's Advanced Practical	**Abbreviation** IMP
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)** depending on teaching offer	**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 examination. This examination includes the assessment of the project results (software, documentation), the project report (5-10 pages), and the presentation (approx. 30 minutes plus discussion). The grade of this examination gives the grade for this module. More details will be given by the lecturer.	

Useful literature	

Application Field

LV-Nr.	**Name** Application Field	**Abbreviation** IAF
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 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’s Thesis

LV-Nr.	**Name** Master’s Thesis	**Abbreviation** IMT
CP 30	**Duration** 6 months	**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 CP. The Master’s thesis includes regular consulting with advisor and the written elaboration.	
Useful literature	will be announced by the advisor	

Master’s Colloquium

LV-Nr.	**Name** Master’s Colloquium	**Abbreviation** IMC
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’s 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’s thesis (recommended)	
Requirements for the assignment of credits and final grade	The module is completed with a graded examination. This examination includes the evaluation of the presentation (approximately 30-60 minutes) and the student’s ability to defend the results of their work in the face of questions and comments (approximately 15-45 minutes). Total time should not exceed 90 minutes. The examination must be passed in order to be awarded the LP. The final grade of the module is determined by the grade of the examination.	
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 information should be considered. As described in the examination regulations, three areas have 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's Advanced Practical can be credited as elective module.

It is possible to credit modules from the Bachelor's degree program Computer Science to the compulsory area of this Masters's degree program on application at the Examination Board, in case basic knowledge required to achieve respective learning objects is missing. For further details please refer to Appendix B of the examination regulations of the Master's degree Data and Computer Science. Please note that such application is mandatory prior to taking the respective course.

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
Advanced Machine Learning (IAML)				●		
Algorithm Engineering (IAE)					●	
Applied Combinatorial Optimization (IACO)					●	
Artificial Intelligence for Programming (IAIP)				●		
Complex Network Analysis (ICNA)					●	
Computational Geometry (ICGeo)	●					
Computer Vision (ICV)	●					
Computer Games (ICS)	●					
Discrete Structures 2 (IDS2)					●	
Distributed and Parallel Algorithms (IDPA)					●	
Fundamentals of Machine Learning (IFML)				●		
Generative Neural Networks for the Sciences (IGNNS)				●		
Geometric Modeling and Animation (IGMA)	●					
Hardware Aware Scientific Computing (IHASC)			●			
Inverse Probleme (IIP)			●			
IT Project Management (IPM)		●				
IT-Sicherheit 2 (IITS2)		●				
Knowledge Management and Decision-Making in Software Engineering (ISWKM)		●				
Machine Learning Essentials (IMLE)				●		
Machine Learning and Physics (MKTP6)				●		
Mining Massive Datasets (IMMD)				●		
Model-Based Time Series Analysis (IMBTSA)				●		
Natural Language Processing with Transformers (INLPT)				●		
Practical Geometry (IPGeo)	●					
Scientific Visualization (ISV)	●					
Software Economics (ISWEco)		●				
Software Evolution (ISWEvol)		●				
Volume Visualization (IVV)	●					
All basic & advanced modules of the MSc Computer Engineering (MScTI)						●

Table 4.1: Module assignment to subject areas.

4.2 Modules from computer science

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

Advanced Machine Learning

LV-Nr.	**Name** Advanced Machine Learning	**Abbreviation** IAML
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 Essentials"' 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, - understand when these methods are called for, what limitations of standard solutions they address, and how they are applied to real-world problems, - 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 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.	
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

LV-Nr.	**Name**	**Abbreviation**
L 1100222000 E 1100222001	Algorithm Engineering	IAE
CP	**Duration**	**Offered**
8	one semester	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, - 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, - are able to select appropriate algorithms to come up with and implement efficient solutions, given a real-world problem, - know realistic machine models and applications, algorithm design, implementation techniques, experimental methodology and can interpret 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 (IADS2)	
Requirements for the assignment of credits and final grade	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.	

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

Applied Combinatorial Optimization

LV-Nr. 1300282205	**Name** Applied Combinatorial Optimization	**Abbreviation** IACO
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 Cannot be combined with Optimization for Machine Learning.
Language English	**Lecturer(s)** Bogdan Savchynskyy	**Examination scheme** 1+1
Learning objectives	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.	

Learning content	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. 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. 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. Fast, accurate enough and learnable algorithms are three aspects we address in this lecture. - Combinatorial problems and their computational complexity: Overview - Linear and integer linear programs and their geometry: Convexity, polyhedra, LP relaxation. - Lifting of variables: Quadratic to linear problem transform, Sherali-Adams hierarchy - Lagrange duality: Subgradient, optimality conditions, relation to LP relaxation, reduced costs. - Systematic exact combinatorial methods: Branching and cutting. - Scalable dual techniques: Non-smooth first order methods, smoothing, primal-dual algorithm. - Greedy algorithms: (Sub-)Optimality, matroids. - Quadratic pseudo-boolean optimization: Algorithms, applications, submodularity. - Scalable primal heuristics: Greedy generation, local search and optimal recombination. Memetic algorithms. - Min-cost-flow: Problem subclasses, theoretical properties and practical algorithms. - Learning parameters of combinatorial problems from training data: Black-box differentiation and recent advances in the literature.
Requirements for participation	recommended are: basic cources: Linear Algebra, Analysis (or, equivalently, Mathematics for computer science) and Algorithms and data structures.
Requirements for the assignment of credits and final grade	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.

Useful literature	Savchynskyy, Bogdan. Discrete graphical models?an optimization perspective. Foundations and Trends® in Computer Graphics and Vision 11.3-4 (2019): 160-429. Boyd, Stephen P., and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004. Korte, Bernhard H. Combinatorial optimization. Berlin: Springer, 2011. Beck, Amir. First-order methods in optimization. Society for Industrial and Applied Mathematics, 2017. Bertsekas, Dimitri P. Nonlinear programming. Journal of the Operational Research Society 48.3 (1997): 334-334. Ahuja, Ravindra K., Thomas L. Magnanti, and James B. Orlin. Network flows. (1988). Papadimitriou, Christos H., and Kenneth Steiglitz. Combinatorial optimization: algorithms and complexity. Courier Corporation, 1998.

Artificial Intelligence for Programming

LV-Nr.	**Name** Artificial Intelligence for Programming	**Abbreviation** IAIP
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 artificial intelligence, in particular knowledge representation, search methods, rule systems, - Basic knowledge about probabilistic models and probabilistic programming, - Knowledge of techniques for code representation and parsing, - Knowledge of techniques for modeling code via neural networks, - Knowledge of basic and advanced methods for program synthesis, - Familiarity with semantic parsing and code summarization, - Familiarity with selected applications of AI for programming, e.g. code-to-code translation, code recommendations, and detection of bugs in code.	
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 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.	

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

LV-Nr.	**Name** Complex Network Analysis	**Abbreviation** ICNA
CP 8	**Duration** one semester	**Offered** every 2nd wintersemester
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. 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 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.
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

LV-Nr.	**Name** Computational Geometry	**Abbreviation** ICGeo
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, - 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, - 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: Algorithmen und Datenstrukturen (IAD)	
Requirements for the assignment of credits and final grade	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.	
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	

Computer Vision

LV-Nr. 1300282207	**Name** Computer Vision	**Abbreviation** ICV
CP	**Duration** one semester	**Offered** every semester
Format Lecture 2 SWS + Exercise 2 SWS	**Workload** 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	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing renaming of Computer Vision: 3D Reconstruction
Language English	**Lecturer(s)** Carsten Rother	**Examination scheme** 1+1
Learning objectives	The students - 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. - understand the principles of image processing and image formation. This can be utilized to design an algorithm for camera calibration. - have studied various techniques for fast object recognition. This can be used to build an object recognition system for e.g. autonomous driving. - understand different approaches for object tracking. - 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. - understand and implement methods that combine machine learning-based methods with classical computer vision-based techniques. - have studied various state-of-the-art computer vision systems and approaches, and are then able to evaluate and classify new systems and approaches.	
Learning content	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 underling 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.	
Requirements for participation	recommended is a basic machine learning background (e.g. Fundamentals of Machine Learning, Advanced Machine Learning or equivalent)	

Requirements for the assignment of credits and final grade	The module is completed with a graded oral or written examination or a graded final report (about 10 pages). 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.
Useful literature	

Computer Games (Game Engine Design)

LV-Nr. 1300132220	**Name** Computer Games (Game Engine Design)	**Abbreviation** ICS
CP 8	**Duration** one semester	**Offered** every summer semester
Format Lecture 3 SWS + Exercise 3 SWS	**Workload** 240 h; thereof 75 h lectures and tutorials 15 h exam preparations 150 h self-study and exercises	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Jürgen Hesser	**Examination scheme** 1+1
Learning objectives	The students - understand the game engine concepts, the decision for specialized class structures, support tools, and the typical architectural elements and are able to apply these concepts in developing an own game engine, - are able to apply and further develop methods for visualizing 3D scenes, perform collision detection and hereby are able to identify the appropriate algorithms, - 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, - will be able to find and apply appropriate techniques for path planning, to improve the found paths to be more realistic, - are able to identify the different concepts of AI in games and develop and apply these techniques for own games. In the exercises, they apply the theoretical concepts and program applications in order to see how to translate concepts into code.	
Learning content	- Overview of the structure and the components of computer games - Architecture of Game Engines - Elements of the Graphics Subengine - Algorithms for Collision Detection - Animation techniques and physics - Path planning and AI	
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 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.	

Useful literature	Gregory et al: Game Engine Architecture Ericson: Real-Time Collision Detection Eberly: Game Physics Millington: Artificial Intelligence for Games

Discrete Structures 2

LV-Nr. L 1100222002 E 1100222003	**Name** Discrete Structures 2	**Abbreviation** IDS2
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: Discrete Structures 1	
Requirements for the assignment of credits and final grade	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.	
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.	

Distributed and Parallel Algorithms

LV-Nr.	**Name**	**Abbreviation**
L 1100222018 E 1100222019	Distributed and Parallel Algorithms	IDPA
CP	**Duration**	**Offered**
8	one semester	every 3rd to 4th semester
Format 4 SWS lecture 2 SWS tutorial, homework assignments	**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 - understand fundamental theoretical and practical concepts of advanced parallel algorithms and data structures, - get to know established methods and algorithms, - are familiar with issues of efficient implementations, - are able to identify/formulate algorithmic problems in/for different application areas where parallel or distributed algorithms are used, - are able to analyse new distributed and parallel algorithms as well as analysing their running time, - and select appropriate algorithms for parallel and distributed applications, - are able to apply parallel and distributed algorithms and data structures to real-world problems, - can objectively assess the quality of the results.	
Learning content	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.	
Requirements for participation	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Lineare Algebra 1	

Requirements for the assignment of credits and final grade	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.
Useful literature	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

Fundamentals of Machine Learning

LV-Nr.	**Name** Fundamentals of Machine Learning	**Abbreviation** IFML
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 Essentials"' 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, - 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 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.	
Useful literature	Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning (2nd edition), Springer, 2009	

Generative Neural Networks for the Sciences

LV-Nr.	**Name**	**Abbreviation**
L 1100222010 E 1100222011	Generative Neural Networks for the Sciences	IGNNS
CP	**Duration**	**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** 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 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.	
Learning content	- 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	
Requirements for participation	recommended: basic knowledge of deep learning and statistics	
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	Kevin Murphy. Probabilistic Machine Learning: Advanced Topics (2023)	

Geometric Modeling and Animation

LV-Nr. L 1100222022 E 1100222023	**Name** Geometric Modeling and Animation	**Abbreviation** IGMA
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 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.
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

LV-Nr.	**Name** Hardware Aware Scientific Computing	**Abbreviation** IHASC
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, - can exploit this parallelism selecting an appropriate programming model, - 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 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.
Useful literature	Frédéric Magoules, François-Xavier Roux, Guillaume Houzeaux: Parallel Scientific Computing, Wiley, 2016, doi: 10.1002/9781118761687

Inverse Problems

LV-Nr. 1300132219	**Name** Inverse Problems	**Abbreviation** IIP
CP 8	**Duration** one semester	**Offered** every summer semester
Format Lecture 2 SWS + Exercise course 2 SWS + Homework	**Workload** 240 h; thereof 60 h lectures and tutorials 15 h exam preparations 165 h self-study and exercises / homework	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Language English	**Lecturer(s)** Jürgen Hesser	**Examination scheme** 1+1
Learning objectives	Students - understand the mathematical properties of inverse problems and are able to demonstrate and show why these problems are difficult to solve, - learn principles of how to solve both deterministic and stochastic problems, - they are able to identify problem settings which request specific deterministic or stochastic approaches and the regularization methods therein, - are able to select an appropriate regularization parameter strategy and understand their differences in particular, - understand how to formulate and apply compressed sensing and deep learning for inverse problems, all principles are presented in selected areas in parameter estimation, - gain the competence in solving complex problems that cannot be dealt with classical techniques, - will be able to adequately evaluate complex experimental measurements.	
Learning content	- Definition of ill-posedness - Deterministic inverse problems, regularization techniques - Tikhonov regularization, data and model resolution matrix, pseudo-inverses - Stochastic inverse problems and Bayes theorem - Regularization parameter selection - Compressed sensing - Deep Learning for Inverse Problems	
Requirements for participation	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Numerische Mathematik	
Requirements for the assignment of credits and final grade	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.	

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

IT Project Management

LV-Nr. 1100222024	**Name** IT Project Management	**Abbreviation** IPM
CP 3	**Duration** one semester	**Offered** every second winter semester
Format lecture + exercise 2 SWS	**Workload** 90 h; thereof 30 h lecture + exercise 15 h preparation for exam 45 h self-study and homework (optionally in groups)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Language English	**Lecturer(s)** Andrea Herrmann	**Examination scheme** 1+1
Learning objectives	The students - are able to plan and control a project, - understand, how projects are embedded into organizations, - have basic knowledge about contractual questions.	
Learning content	- Project planning, project organization - Cost estimation - Project offer, contract, negotiations - Pprocess models - Risk management - Controlling - IT contract laws - Change management - Time management - Project closure - Distributed software engineering	
Requirements for participation	none	
Requirements for the assignment of credits and final grade	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.	
Useful literature	PMI (Project Management Institute): A Guide to the Project Management Body of Knowledge (PM BOK ® Guide), 6th Edition, 2017	

IT-Sicherheit 2

LV-Nr. V 1100222004 Ü 1100222005	**Name** IT-Sicherheit 2	**Abbreviation** IITS2, ITSec2
CP 6	**Duration** ein Semester	**Offered** unregelmäßig
Format Vorlesung 2 SWS + Übung 2 SWS	**Workload** 180 h; davon 60 h Präsenzstudium 15 h Prüfungsvorbereitung 105 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	**Availability** nicht kombinierbar mit Modul IT-Sicherheit für 8 LP M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Language Deutsch	**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 erweiterte Kenntnisse über die Sicherung großer Netzwerke und der Kommunikationsinfrastruktur (Routing, Namensauflösung, Internet-Firewalls, Intrusion Detection Systeme), - erwerben vertiefte Kompetenzen zur Detektion von Cyberangriffen, - erwerben grundlegende Kompetenzen im Bereich Penetration Testing. Langfristiges Ausbildungsziel: Einsatz- und Beschäftigungsfähigkeit in der Breite des Arbeitsfeldes IT-Sicherheit.	
Learning content	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: - Sicherheitsmodelle und Bewertungskriterien - Authentifikationsverfahren - Schutz von Kommunikationsinfrastruktur; Netzsicherheit - Digitale Identität - Software-Exploitation - Penetration Testing - Zero Trust Security Mit Hilfe von virtuellen Maschinen in einem geschützten Bereich werden klassische Angriffs- und Schutzszenarien praktisch untersucht.	

Requirements for participation	empfohlen ist: IT-Sicherheit 1 (IITS1)
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	

Knowledge Management and Decision-Making in Software Engineering

LV-Nr. 1100222016	**Name** Knowledge Management and Decision-Making in Software Engineering	**Abbreviation** ISWKM
CP 3	**Duration** one semester	**Offered** every second winter semester
Format lecture + exercise 2 SWS	**Workload** 90 h; thereof: 30 h lecture + exercise 15 h preparation for exam 45 h self-study and homework (optionally in groups)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Language English	**Lecturer(s)** Andrea Herrmann	**Examination scheme** 1+1
Learning objectives	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.	
Learning content	- Knowledge management - Onthologies and Grounded Theory - Reverse engineering, code metrics - Learning organization - Storytelling - Decision-making and decision theory - Management decisions, business case - Risk management - Requirements prioritization - Decision-making in design: ATAM, SAAM, CBAM - Decision-making under uncertainty - Mathematical Economics - Decision-making with several parties: Harvard concept, negotiations, Game Theory - Decision Traps and Biases - Ethical decisions and machine ethics	
Requirements for participation	recommended are: Einführung in Software Engineering (module ISW) or comparable competences	
Requirements for the assignment of credits and final grade	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.	

Useful literature	Raiffa, Howard; Richardson, John; Metcalfe, David: Negotiation analysis - the science and art of collaborative decision making, Belknap, Cambridge, 2002 or 2007

Machine Learning Essentials

LV-Nr.	**Name**	**Abbreviation**
L 1100222006 E 1100222007	Machine Learning Essentials	IMLE
CP	**Duration**	**Offered**
8	one semester	in (irregular) alternation with "'Fundamentals of Machine Learning"' and "'Advanced 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** 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
Language English	**Lecturer(s)** Ullrich Köthe	**Examination scheme** 1+1
Learning objectives	The 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. - 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"' and "'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	This is the retitled "'Machine Learning"' module. 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.
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

LV-Nr. L 1100222012 E 1100222013	**Name** Mining Massive Datasets	**Abbreviation** IMMD
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	The students - know selected approaches and programming paradigms of parallel data processing, - know how to use tools for parallel data processing (among others Apache Hadoop and Spark), - are familiar with application domains of big data analysis, - know methods of parallel pre-processing of data, - know methods like classification, regression, clustering and their parallel implementations, - know about 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 classification, regression, clustering and evaluation of the results - Parallel algorithms for data analysis and their implementations - Theory and practice of scalability and tuning of frameworks	
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 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.	

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)

Model-Based Time Series Analysis

LV-Nr.	**Name**	**Abbreviation**
L 1100222028 E 1100222029	Model-Based Time Series Analysis	IMBTSA
CP	**Duration**	**Offered**
8	one semester	irregular
Format Lecture 4 SWS + Exercise course 2 SWS	**Workload** 240h, thereof 60h lecture, 30h exercises, 126h self-study and working on assignments (optionally in groups), 24h exam preparation	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing Renaming of Time Series Analysis With Applications to Cognitive Science
Language English	**Lecturer(s)** Georgia Koppe	**Examination scheme** 1+1
Learning objectives	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.	
Learning content	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	
Requirements for participation	recommended prior knowledge in basic calculus, statistics, and linear algebra	
Requirements for the assignment of credits and final grade	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.	

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

Natural Language Processing with Transformers

LV-Nr.	**Name** Natural Language Processing with Transformers	**Abbreviation** INLPT
CP	**Duration**	**Offered** every 2nd winter semester
Format Lecture 2 h + Exercise course 2 h	**Workload** 180 h; thereof 60 h lecture 120 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 - fully understand the principles and methods underlying word embedding approaches, - are familiar with traditional sequence-to-sequence machine learning methods, - can describe the key concepts and techniques underlying attention mechanisms and different transformer architectures, - understanding training and fine-tuning approaches to improve the performance of different transformer architectures for different downstream NLP tasks, - know the key methods and architectural components for building QA and text summarization pipelines, - can build and deploy QA and text summarization pipelines using common software frameworks, - know key metrics in evaluating transformer architectures for different applications, - can implement diverse transformer-based NLP applications using common Python frameworks and libraries, - can deploy transformer-based NLP applications through Web interfaces.	
Learning content	- Word embeddings (review of simple neural network architectures and concepts) - Sequence-to-sequence models (Recurrent Neural Networks, LSTM, GRU) - Attention mechanism - Transformer components (encoder, decoder) and common transformer architectures (BERT, GPT, T5) - Training and fine-tuning transformers, including zero- and few-shot learning - Text summarization approaches - Question answering and building a QA pipeline - Transformer architectures for conversational AI - Programming and model frameworks such as Huggingface, LangChain, OpenAI and (cloud-based) vector databases	

Requirements for participation	Recommended courses: Data Science for Text Analytics (IDSTA), Foundations of Machine Learning (IML) 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
Requirements for the assignment of credits and final grade	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.
Useful literature	For the different topics, several research papers will be provided to students via the Moodle platform. The following textbooks are useful but not required: Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Natural Language Processing with Transformers, 2022 (revised edition) Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft) Furthermore, during the course of this lecture, several papers covering topics discussed in class will be provided.

Practical Geometry

LV-Nr.	**Name** Practical Geometry	**Abbreviation** IPGeo
CP 4	**Duration** one semester	**Offered** irregularly
Format Lecture 2 SWS + Exercise course 1 SWS	**Workload** 120h; thereof 45 h lecture 60 h self-study and working on assignments 15h preparation for exam	**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 - understand basic geometric concepts for data analysis as well as efficient point search and further processing of measurement data - confidently handle projections and descriptions beyond the three-dimensional world of experience, - calculate geometric invariants, distances, curvatures from measurement data, reconstructed and generated surfaces.	
Learning content	Basic areas of geometry with relevance in computer graphics, image processing, pattern recognition, computer vision and geometric modeling. (i) Analytic geometry: operations on vector spaces with appropriate coordinates and mappings (affine mappings, collinearities), geometric fitting of point clouds to linear structures or planes from error-prone measurement data (ii) Projective geometry: central projection and inverse reconstruction of 3D objects from planar images (computer vision, geodesy), differences between B-spline curves and surfaces and the class of NURBS, freeform geometries in CAD systems (iii) Differential geometry: parameter representations in geometric data processing, implicit representations (level sets), estimation of invariants from discrete data (triangulations, point clouds).	
Requirements for participation	recommended are: linear algebra, computational geometry and any 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 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.	
Useful literature	Geometrie für Informatiker, Script TU Vienna 2004, Helmut Pottmann, current publications	

Projektseminar Biomedizinische Bildanalyse

LV-Nr.	**Name** Projektseminar Biomedizinische Bildanalyse	**Abbreviation** IPBB
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** 1+1
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

LV-Nr. L 1100222020 E 1100222021	**Name** Scientific Visualization	**Abbreviation** ISV
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, - understand the mathematical fundamentals, data structures, and implementation aspects - get to know schemes for interpolation and integration, mapping for scalar, vector, and tensor fields, and derived approaches, - understand approaches for direct and indirect volume rendering, feature extraction, and topology-based analysis, - are able to apply these concepts to real-world problems using existing software packages, and develop small programs using visualization libraries.	
Learning content	- 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 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.	

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

Software Economics

LV-Nr.	Name	Abbreviation
1100222026	Software Economics	ISWEco
CP	**Duration**	**Offered** irregularly
Format Lecture 2 SWS	**Workload** 90 hours; thereof 30 hours lecture 35 hours individual processing / self-study 25 hours preparation for exam (in groups possible / recommended)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Language English	**Lecturer(s)** Eckhart von Hahn	**Examination scheme** 1+1
Learning objectives	After a successful participation in the lecture the students can - roughly determine the price and licensing of an already created software, - plan and initiate marketing activities for software and software-related services / products, - roughly understand the balance sheet and profit-and-loss statement of a software manufacturer, - assess the value of a software with its various components, from the perspective of the manufacturer as well as from the perspective of the user, - plan price negotiations for software projects. The students knows afterwards - the basics of cost and performance accounting (as far as it is relevant for software creation), - the different types of (legal) contracts that are used in the area of software creation, - the most important negotiation strategies when negotiating software contracts, - legal aspects in the area of IT crime, - as well as the relevance of the lecture topics in the practice of industrial software creation.	

Learning content	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: - Disambiguation of terms - Economic aspects during the planning and creation phase of the software lifecycle - Economic aspects during the value assessment phase - Economic aspects during the value transfer phase - Accounting aspects - Maintaining the value of software
Requirements for participation	recommended are knowledge and skills taught in the module Introduction to Software Engineering (ISW)
Requirements for the assignment of credits and final grade	The module is concluded with a graded exam - oral or written. Details are provided at the beginning of the lecture.
Useful literature	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

Software Evolution

LV-Nr. 1100222014	**Name** Software Evolution	**Abbreviation** ISWEvol
CP 3	**Duration** one semester	**Offered** irregularly
Format Lecture 2 SWS	**Workload** 90h; thereof 30h lecture 35h individual processing / self-study 25h preparation for exam (in groups possible / recommended)	**Availability** M.Sc. Angewandte Informatik M.Sc. Data and Computer Science
Language English	**Lecturer(s)** Eckhart von Hahn	**Examination scheme** 1+1
Learning objectives	After the successful participation in the lecture the students can - create a maintenance concept for an existing software, - plan a software reengineering project from a technical / functional perspective, - develop a framework to enable a sustainable software development during the initial creation phase. The students know afterwards - the typology of software maintenance and the management of troubleshooting, - the classical array of software revitalization techniques (e.g. refactoring), - the difference and the challenges of progressive development versus the initial creation of software and on which aspects you have to pay particular attention – through the lense of the provider of a software as well as the user of a software, - in general the relevance of the topic for the industrial engineering practice.	
Learning content	This module intends to convey the concepts for a successful software engineering lifecycle after its initial creation. 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: - Disambiguation of terms - Software Maintenance - Software Reengineering - Progressive Software Development / Software Evolution in particular and its management - Software Migration	
Requirements for participation	recommended are knowledge and skills taught in the module Introduction to Software Engineering (ISW)	

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

Volume Visualization

LV-Nr. 1300132218	**Name** Volume Visualization	**Abbreviation** IVV
CP 8	**Duration** one semester	**Offered** every summer semester
Format Lecture 4 SWS + Exercise course 3 SWS	**Workload** 240 h; thereof 75 h lecture 15 h preparation for exam 150 h self-study and working on assignments/projects (optionally in groups)	**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, - 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 volume representations and their efficient realizations - Theory of volume rendering and their different realizations - Acceleration 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 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.	
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	

4.3 Modules from BSc/MSc Mathematics

4.3.1 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)

4.3.2 Master of Mathematics

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

Basic Module Numerics and Optimization (MM15)

- Nonlinear Optimization
- Uncertainty Quantification 1

Specialization Module Numerics and Optimization (MM35)

- Computational Fluid Dynamics
- Fundamentals of Computational Environmental Physics
- Mathematical Methods of Image and Pattern Analysis II

From the supplementary modules

- Computability and Complexity I
- Computability and Complexity II

4.4 Modules from MSc Physics

From the MSc Physics the module *Machine Learning and Physics (MKTP6)* can be credited in the MSc Data and Computer Science. The description of the module can be found in the current module handbook of the MSc Physics.

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.

5 Interdisciplinary competencies

Interdisciplinary competencies (in German *Übergreifende Kompetenzen ÜK*) refer to study contents, key competencies and additional qualifications that go beyond subject-specific knowledge and convey personality and job-related competencies that are essential in today's professional life (in and outside of research). A maximum of 6 credit points can be earned in the area of interdisciplinary competencies (ÜK). There are various choices available (some module descriptions follow on the next pages). Within the framework of the ÜK, courses from the university's range of courses that do not belong to the computer science program or the application area can be accepted. This includes language courses, but not courses of the Heidelberg University Computer Center (URZ). In this case, the credit points of the courses are transferred (especially for language courses). Courses offered by the Career Service in the area of ÜK can be recognized; in this case, it is essential to consult with the Examination Office beforehand. Furthermore, irregular offers of the faculty marked as ÜK can be taken.
From the Master of Computer Engineering the module *Entrepreneurship* can be chosen, it is recognized with 6 LP. For the module description please refer to the module handbook of the Master Computer Engineering course of studies. The module *Tools* can not be chosen.

Einführung in das Textsatzsystem LaTeX

LV-Nr.	**Name** Einführung in das Textsatzsystem LaTeX	**Abbreviation** ILat
CP 2 ÜK	**Duration** ein Semester	**Offered** unregelmäßig
Format Praktikum 2 SWS	**Workload** 60 h; davon 30 h Präsenzstudium 15 h praktische Übung am Rechner 15 h Hausaufgaben	**Availability** B.Sc. Angewandte Informatik B.Sc. Informatik B.Sc. Mathematik M.Sc. Scientific Computing
Language Deutsch	**Lecturer(s)** wechselnd	**Examination scheme** 1+1
Learning objectives	Nachdem Studierende die Veranstaltung besucht haben, können sie - ein TeX-System installieren und einrichten, - LaTeX-Dokumente mit komplexer Struktur erstellen und bearbeiten, - gängige Fehler in LaTeX-Dokumenten identifizieren und beheben, - LaTeX-Makros programmieren, - LaTeX-Umgebungen mit verschiedenen Paketen aufsetzen.	
Learning content	Der Kurs gibt eine Einführung in das Satzsystem LaTeX und vermittelt grundlegende typographische Kenntnisse. Ziel des Kurses ist es, längere und komplexe Dokumente (z. B. Bachelor- und Masterarbeiten sowie Dissertationen) eigenständig in hoher Qualität zu entwickeln, ohne auf die Probleme zu stoßen, die ein komplexes System wie LaTeX dem Anfänger bereitet. Es werden weiterhin auch moderne Konzepte und Entwicklungen von LaTeX vorgestellt, die dem Anwender interessante und hilfreiche Tools zur Verfügung stellen. Behandelt werden u.a. - allgemeine Formatierung, Pakete Schriften, - Gleitobjekte: Bilder, Tabellen, - Verzeichnisse, - Mathematiksatz, - mehrsprachige Dokumente, - Präsentationen, - Diagramme, - Typographische Feinheiten, - Professionelle Briefe, Lebenslauf.	
Requirements for participation	none	
Requirements for the assignment of credits and final grade	Die Details werden zu Beginn der Lehrveranstaltung bekannt gegeben.	

Useful literature	

Industrial Internship

LV-Nr.	**Name** Industrial Internship	**Abbreviation** IInd
CP 1 ÜK pro 30 h	**Duration**	**Offered**
Format Working in an industrial company	**Workload** 120 hours; thereof at least 110 hours presence in the company 10 hours to write the report	**Availability** B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Data and Computer Science
Language	**Lecturer(s)** Chairperson of the examination board	**Examination scheme** 1+1
Learning objectives	Learning and application of methods and tools in hardware and/or software development in an industrial context.	
Learning content	The industrial internship is intended to impart project-related application of IT methods in hardware and/or software development. Ideally, the internship should be embedded in a process (e.g., in software development) in which the task is clearly specified by the company and the solution is worked out during the internship (in a team). Tasks, such as pure software installation, hardware installation, operating system updates or customer help desk, do not count as internship content.	
Requirements for participation	Before starting an industrial internship, it should be clarified with the chairperson of the examination board of the degree course whether and to what extent the planned content of the internship can be credited.	
Requirements for the assignment of credits and final grade	The awarding of the CP does not only depend on the duration (time commitment) of the internship, but also on the content. For this purpose, an approx. 6-page, well-structured written report (PDF, A4, 11 pt, max. 1.5 line spacing) on the activities carried out, including the tasks and results, must be provided. A letter on the type and duration of the internship, signed by the supervisor in the company, must be attached to the report. The report is graded as pass or fail.	
Useful literature		

Education through Summer School, Holiday Course, or Conference

LV-Nr.	**Name** Education through Summer School, Holiday Course, or Conference	**Abbreviation** IBil
CP 1 ÜK pro 30 h	**Duration**	**Offered**
Format Participation in a computer science event with content that is not taught in the computer science degree course	**Workload** at least 30 hours presence at the event	**Availability** B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language	**Lecturer(s)** Chairperson of the examination board	**Examination scheme** 1+1
Learning objectives	Experience with subject-specific content that goes beyond the studies, including its intensive discussions.	
Learning content		
Requirements for participation		
Requirements for the assignment of credits and final grade	The module is concluded with an ungraded exam. This exam includes a written report on the event and the experiences gained (approx. 1 page per CP). This report must be passed in order for the CP to be awarded.	
Useful literature		

Study Abroad

LV-Nr.	**Name** Study Abroad	**Abbreviation** IAus
CP 4 ÜK für 3 Zeitmonate	**Duration** 3 Monate	**Offered**
Format Studies outside of Germany	**Workload** 160 hours; thereof 120h settling into the foreign study context 40h reflection and reporting	**Availability** B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language	**Lecturer(s)** Chairperson of the examination board	**Examination scheme** 1+1
Learning objectives	Experience with everyday study life in a different country	
Learning content		
Requirements for participation		
Requirements for the assignment of credits and final grade	The module is concluded with an ungraded exam. This exam includes an approximately 4-page written report on the studies and the experiences made. This report must be passed in order for the CP to be awarded.	
Useful literature		