Module Handbook

International Master
"Scientific Computing"

Universität Heidelberg
Fakultät für Mathematik und Informatik

Version as of 15.07.2026 corresponding to examination regulations of 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

date of begin: 11.03.2009

location of study: Heidelberg

number of places: no limitation

fee: according to general regulations of Heidelberg University

Contents

1 Qualification objectives, profile, and particularities of the degree programme **4**
- 1.1 Preamble - Qualification objectives of Heidelberg University . . . 4
- 1.2 Profile of the degree programme . . . 4
- 1.3 Subject-specific qualification objectives of the degree programme . . . 4
- 1.4 Generic qualification objectives of the degree programme . . . 5
- 1.5 Graduates of the degree programme may enter any of the following professions . . . 5
- 1.6 Particularities of the degree programme and module descriptions . . . 5
  - 1.6.1 Reason for cumulative examinations . . . 5
  - 1.6.2 Reason for modules with fewer than 5 credits . . . 5
  - 1.6.3 Description of the teaching and learning forms . . . 6
  - 1.6.4 Modalities for exams . . . 6

2 Typical Plan of Study and Mobility **7**
- 2.1 Typical Plan of Study . . . 7
- 2.2 Mobility window . . . 7

3 Elective compulsory Modules **8**
- 3.1 Courses in Computer Science . . . 8
  - Algorithm Engineering . . . 9
  - Algorithmic Randomness and Computable Analysis . . . 11
  - Applied Combinatorial Optimization . . . 12
  - Automated Planning . . . 15
  - Complex Network Analysis . . . 16
  - Computational Geometry . . . 18
  - Computer Games (Game Engine Design) . . . 19
  - Computer Vision . . . 21
  - Discrete Structures 1 . . . 23
  - Discrete Structures 2 . . . 24
  - Distributed and Parallel Algorithms . . . 25
  - Generative Neural Networks for the Sciences . . . 27
  - Geometric Modeling and Animation . . . 28
  - Hardware Aware Scientific Computing . . . 30
  - Inverse Problems . . . 32
  - Machine Learning Essentials . . . 34
  - Mining Massive Datasets . . . 35
  - Model-Based Time Series Analysis . . . 37
  - Practical Geometry . . . 39
  - Reinforcement Learning . . . 40
  - Robotics and Machine Intelligence . . . 42
  - Scientific Visualization . . . 44
  - Volume Visualization . . . 46
  - Modules from the MSc Technische Informatik . . . 47

3.2 Courses in Mathematics . . . 47
Grundmodul Angewandte Analysis und Modellierung . . . 48
Basic course Numerical Analysis and Optimization . . . 51
Grundmodul Statistik und Wahrscheinlichkeitsrechnung . . . 54
Aufbaumodul Angewandte Analysis und Modellierung . . . 57
Advanced course numerical analysis and optimization . . . 59
Aufbaumodul Statistik und Wahrscheinlichkeitsrechnung . . . 62
Spezialisierungsmodul Angewandte Analysis und Modellierung . . . 64
Special topics of numerical analysis and optimization . . . 65
Spezialisierungsmodul Statistik und Wahrscheinlichkeitsrechnung . . . 66
Applied Mathematics in Biology: Single-cell omics . . . 67

4 General Modules **69**
Master Seminar (Mathematics) . . . 70
Master Programming Practical . . . 71
Master Thesis . . . 73
Master Thesis Presentation . . . 74
Field of Application . . . 75
Lab internship in field of application . . . 76

5 Key Competence Program **77**
Study Abroad . . . 78
Industrial Internship for Scientific Computing . . . 79
Education through Summer School, Holiday Course, or Conference . . . 80
Introduction to LaTeX . . . 81
IT Project Management . . . 83
Software Economics . . . 84

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

1.1 Preamble - Qualification objectives of Heidelberg University

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

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

1.2 Profile of the degree programme

The international master program "Scientific Computing" aims at expanding proficiency in mathematics and computer science with a particular focus on the interplay of the two towards topics relevant to cutting edge research at Heidelberg University.

1.3 Subject-specific qualification objectives of the degree programme

Graduates of the master program are able to apply their in-depth knowledge in one or more areas of Applied Mathematics and Computer Science such as machine learning and data analysis, numerical modelling, simulation and optimization, or scientific visualization and computer graphics. They are able to analyse and decompose complex real-world problems into manageable components and develop mathematical and computational approaches for them.

Graduates are trained to select and efficiently implement appropriate computational methods of mathematics and computer science and, if necessary, independently tailor or advance them to meet their needs. Graduates are competent in identifying the adequate computational resources to simulate said models. They are able to transfer insights gained back to the original problem setting and draw meaningful conclusions.

1.4 Generic qualification objectives of the degree programme

Graduates of the programme are expected to possess the following competences of interdisciplinary nature:

- Graduates have acquired wide-ranging problem-solving skills, are proficient in their application and competent in applying them to new, unfamiliar situations.
- They are competent in collecting independent information, to make judgments and to acquire knowledge in their field as well as from related disciplines. In particular, they are capable of procuring and interpreting scientific literature and of evaluating alternative solutions in their areas of specialization.
- Graduates are able to communicate and present their results and conclusions orally and in writing to experts and non-experts and to engage in disciplinary and interdisciplinary scientific discussions.
- 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.
- Graduates have the competence to work in disciplinary and interdisciplinary teams, to take on team leadership and communicate effectively within the team.
- They apply time-management and organizational skills confidently.

1.5 Graduates of the degree programme may enter any of the following professions

Graduates are well versed to enter any profession requiring advanced problem-solving, computational, and analytic skills. This includes careers in a wide range of disciplines in industry, academia and the public sector, such as computational scientist, mathematical modeller, data scientist, computational engineer, simulation specialist, software engineer, project manager in research and development, academic researcher, scientific manager, and others.

1.6 Particularities of the degree programme and module descriptions

1.6.1 Reason for cumulative examinations

Several modules in the degree programme are aiming to assess both practical skills as well as acquired theoretical knowledge, which in some cases warrants cumulative assessments made up of two or more components that are able to check all aspects.

1.6.2 Reason for modules with fewer than 5 credits

There are some modules in this programme with less than 5 credits. These modules are self-contained units of study in terms of content which do not justify more credits and cannot reasonably be combined with other modules.

1.6.3 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, written elaboration of the content.
- **Practical:** Project work on the basis of a programming task, independent development of software including documentation, preparation of a project report and a lecture, giving a lecture on the presentation of the software (20-30 minutes).

1.6.4 Modalities for exams

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

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

Rules for awarding the CP: The CP are awarded if the final examination is passed. The details of the final examination are described in the individual module descriptions. Many modules include exercise sessions using various models. The requirements for admission to the final examination are determined by the lecturer at the beginning of the course and announced in writing.

Examination scheme: This part of the module description specifies the number of attempts which are allowed to pass the module, according to the examination regulations. Once an examination 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: Two exam periods have been designated for the written exams at the end of each semester. The first exam period lasts three weeks and consists of the last week of the lecture period and the first two weeks of the lecture-free period. The second exam period lasts three weeks and consists of the last two weeks of the lecture-free period and the first week of the lecture period. In exceptional cases, exams may take place outside of these exam periods; lecturers must announce such exceptions at the beginning of the course.
For oral exams, the dates are set by the lecturers.

Examination dates: For modules offered once a year or less frequently, and for which the module exam is a written exam, two exam dates are always offered following the module. These exams are scheduled during the first and second exam periods, respectively. Students choose which of the offered exam dates they wish to take.
For smaller (irregularly offered) modules, exceptions to this rule may be made. The lecturers must announce any such exceptions at the beginning of the course.
For modules that are offered every semester and have a written exam as the module exam, there is only one exam date following the module, within the first exam period. Students who fail or withdraw for a valid reason may retake the exam in the following semester.

2 Typical Plan of Study and Mobility

2.1 Typical Plan of Study

1st year:		
	Elective compulsory modules in Mathematics	16 CP
	Elective compulsory modules in Computer Science	16 CP
	Elective compulsory modules	6 CP
	Master Seminar (Mathematics)	6 CP
	Field of Application	16 CP
sum		**60 CP**
2nd year:		
	Elective compulsory modules	10 CP
	Key Competence Program	6 CP
	Master Programming Practical	8 CP
	Master Thesis	30 CP
	Master Thesis Presentation	6 CP
sum		**60 CP**
total:		**120 CP**

2.2 Mobility window

The master's program Scientific Computing offers a large variety of possibilities for student mobility due to the strong international research collaboration of the scientists involved. Based on a solid methodological education background, the third term of the program is particularly suitable for spending a term abroad, but the study program also actively supports individual mobility concepts.

Students are advised to plan a mobility measure well in advance in cooperation with the counseling office of the program. Applications for support by the ERASMUS program and the 4EU+ alliance take planning phase of up to 10 months.

Spending a term abroad opens up the possibility to learn about different research topics and might well lead to a master thesis under the joint supervision of scientists from both the host and the home university. Short-term mobilities such as the participation in summer schools or single compact courses within the framework of the program are also encouraged and credits from such activities can be recognized based on the regular recognition of external modules.

3 Elective compulsory Modules

For the elective compulsory modules a total of 48 CP must be earned. At least 16 CP must come from mathematics and 16 CP from computer science. These two areas with the modules are listed in the sections below.

3.1 Courses in Computer Science

In this chapter all module descriptions for the modules in computer science are listed. At the end there is a list of modules from the MSc Technische Informatik which are also allowed to choose as modules in computer science.

For the elective compulsory modules in Computer Science 16 CP must be recieved.

Algorithm Engineering

LV-Nr. L 1100222000 E 1100222001	**Name** Algorithm Engineering	**Abbreviation** IAE
CP 8	**Duration** one semester	**Offered** every summer semester
Format Lecture 4 SWS + Exercise course 2 SWS	**Workload** 240h; thereof 90h lectures and tutorials, 15h exam preparations, 135h lecture wrap-up and homework	**Availability** M.Sc. 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 die Informatik 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

Algorithmic Randomness and Computable Analysis

LV-Nr. L 1100222040	**Name** Algorithmic Randomness and Computable Analysis	**Abbreviation** IARCA
CP	**Duration**	**Offered** irregular
Format Lecture 2 SWS	**Workload** 120h; thereof 30h lectures, 15h exam preparations, 75h lecture wrap-up and homework	**Availability** M.Sc. Data and Computer Science M.Sc. Mathematik M.Sc. Scientific Computing
Language English	**Lecturer(s)** Ivan Titov	**Examination scheme** 1+1
Learning objectives	The students will learn basics of computability theory and information theory and obtain basic knowledge of algorithmic randomness and computable analysis; after passing the course, they will be able to differentiate between various randomness concepts and relate them to notions of approximability of real numbers.	
Learning content	Turing machines and Computability. Various notions of Kolmogorov complexity. The main approaches to characterize effective randomness: typicality, unpredictability and incompressibility. Effective approximability of real numbers and randomness of reals. Relative randomness and reducibilities. Basics of computable analysis.	
Requirements for participation	recommended are: basic knowledge of Turing machines and computability theory	
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.	
Useful literature	Rodney G. Downey and Denis R. Hirschfeldt, Algorithmic Randomness and Complexity. Theory and Applications of Computability, Springer, New York, 2010. MR 2732288 Alexander Shen, Vladimir Andreevich Uspensky, Nikolay Vereshchagin. Kolmogorov Complexity and Algorithmic Randomness. AMS, 2017, ISBN 9781470431822 Klaus Weihrauch. Computable Analysis: An Introduction. Springer-Verlag, Berlin, Heidelberg, 2000. DOI:10.1007/978-3-642-56999-9	

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. 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.

Automated Planning

LV-Nr. L 1100222034 E 1100222035	**Name** Automated Planning	**Abbreviation** IAuPl
CP	**Duration**	**Offered** irregular
Format Lecture 2 SWS + Exercise course 2 SWS	**Workload** 180 h; thereof 60 h lectures and tutorials, 15 h preparation for exam, 105 h lecture wrap-up and homework	**Availability** M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Daniel Gnad	**Examination scheme** 1+1
Learning objectives	Students can: - model planning problems in a common formalism - implement common search techniques and heuristics - compare planning methods - apply planning algorithms to solve planning problems - analyze the performance of planning algorithms - discuss the complexity of planning problems	
Learning content	- planning formalisms (STRIPS, FDR, PDDL) - structural analysis of planning problems - planning as heuristic search - common heuristic functions (delete relaxation, landmarks, abstractions) - complexity of solving planning problems - search space reduction techniques - symbolic planning (planning as SAT)	
Requirements for participation	recommended are: Algorithmen und Datenstrukturen, Artificial Intelligence	
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		

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. 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), Lineare Algebra I (MA4), Einführung in die Wahrscheinlichkeitstheorie und Statistik (MA 8)	

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. 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 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 2 SWS	**Workload** 240 h; thereof 75 h lectures and tutorials 15 h exam preparations 150 h self-study and exercises	**Availability** 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

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. Data and Computer Science M.Sc. Scientific Computing
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. Machine Learning Essentials)	

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	

Discrete Structures 1

LV-Nr. L 1100212006 E 1100212007	**Name** Discrete Structures 1	**Abbreviation** IDS1
CP 8	**Duration** one semester	**Offered** every winter 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** B.Sc. Informatik B.Sc. Mathematik
Language English	**Lecturer(s)** Felix Joos	**Examination scheme** 1+1 (im BSc Mathematik 1+2)
Learning objectives	Students - understand several basic graph parameters and the central theorems in these areas, - can solve easy problems involving discussed topics, - can describe graph algorithms computing discussed graph parameters, - know how to use graphs and graph parameters to model real world problems.	
Learning content	- Introduction to graph theory terminology, - Matchings in graphs and hypergraphs, - Graph connectivity, - Planar graphs, - Graph Colouring, - Hamilton Cycles, - Ramsey Theory, - Random graphs, - Algebraic Graph constructions (Cayley graphs, Kneser graphs,...), - Algorithms computing discussed graph parameters.	
Requirements for participation	recommended are: Einführung in die Praktische Informatik (IPI), Mathematik für die Informatik 1 (IMI1) or Lineare Algebra 1 (MA4), Mathematik für die Informatik 2 (IMI2) or Analysis 1 (MA1)	
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.	

Discrete Structures 2

LV-Nr. L 1100222002 E 1100222003	**Name** Discrete Structures 2	**Abbreviation** IDS2
CP 8	**Duration** one semester	**Offered** every 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. 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. L 1100222018 E 1100222019	**Name** Distributed and Parallel Algorithms	**Abbreviation** IDPA
CP 8	**Duration** one semester	**Offered** every 3rd to 4th 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. 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

Generative Neural Networks for the Sciences

LV-Nr. L 1100222010 E 1100222011	**Name** Generative Neural Networks for the Sciences	**Abbreviation** 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. 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. 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. L 1100222030 E 1100222031	**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. 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. Data and Computer Science M.Sc. Scientific Computing
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

Machine Learning Essentials

LV-Nr. L 1100222006 E 1100222007	**Name** Machine Learning Essentials	**Abbreviation** IMLE
CP 8	**Duration** one semester	**Offered** irregular
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. 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	Classification (linear and quadratic discriminant analysis, neural networks, linear and kernelized support vector machines, decision trees and random forests), least squares and regularized regression, Gaussian processes, unsupervised learning (density estimation, cluster analysis, Gaussian mixture models and expectation maximization, principal component analysis, bilinear decompositions), directed probabilistic graphical models, optimization for machine learning, structured learning	
Requirements for participation	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
Requirements for the assignment of credits and final grade	The module is completed with a graded written 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. 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. L 1100222028 E 1100222029	**Name** Model-Based Time Series Analysis	**Abbreviation** IMBTSA
CP	**Duration**	**Offered** 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. 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.

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. 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	

Reinforcement Learning

LV-Nr. L 1100222038 E 1100222039	**Name** Reinforcement Learning	**Abbreviation** IRL
CP	**Duration**	**Offered** irregular
Format Lecture 2 SWS + Exercise course 2 SWS	**Workload** 180 h; thereof 60 h lectures and tutorials, 15 h preparation for exam, 105 h lecture wrap-up and homework	**Availability** M.Sc. Data and Computer Science M.Sc. Scientific Computing
Language English	**Lecturer(s)** Georgia Koppe	**Examination scheme** 1+1
Learning objectives	Students - understand the foundations of reinforcement learning embedded in the Markov decision process (MDP) framework - can derive and analyze value functions, policies, and dynamic programming methods - understand Monte Carlo and temporal-difference prediction and control - can implement core reinforcement learning algorithms in self-written code - evaluate trade-offs between model-free and model-based methods and relate them to cognitive and computational models of learning and decision-making	
Learning content	- Core concepts of reinforcement learning: reward signals, returns, Markov property - The Markov decision process formalism: states, actions, transitions, and value functions - Dynamic programming methods: policy evaluation, policy iteration, value iteration - Monte Carlo and temporal-difference learning: TD(0), SARSA, Q-learning - Planning and model-based reinforcement learning: Dyna, prioritized sweeping - Function approximation and generalization: linear function approximation, stochastic gradient descent - Policy gradient methods and actor-critic architectures - Exploration versus exploitation and information-seeking behavior - Hierarchical and multi-step methods - Applications in artificial intelligence, control, and cognitive modeling - Evaluation of learning agents: convergence diagnostics, sample complexity, and performance metrics	
Requirements for participation	recommended are: solid knowledge of basic calculus, linear algebra, and statistics, as well as Python programming skills	

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 at the beginning of this course.
Useful literature	Sutton, R.S. and Barto, A.G. (2018). Reinforcement Learning: A Introduction, 2nd edition. MIT Press.

Robotics and Machine Intelligence

LV-Nr. L 1100222042 E 1100222043	**Name** Robotics and Machine Intelligence	**Abbreviation** IRMI
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), including practical exercises on robot hardware	**Availability** M.Sc. Data and Computer Science
Language English	**Lecturer(s)** Artur Andrzejak	**Examination scheme** 1+1
Learning objectives	The students: - know the mathematical foundations of robot manipulators, including rigid-body transformations, forward and inverse kinematics, Jacobians, and trajectory generation, - understand basic robot dynamics and control and their limitations, motivating learning-based methods, - know deep learning methods for robot perception, in particular 2D and 3D computer vision, - know the main paradigms of learning robot behavior, in particular imitation learning and reinforcement learning, and can apply suitable algorithms to a given task, - understand the role of simulation, sim-to-real transfer, and safety when deploying learned policies on real robots, - know modern foundation-model approaches to robotics, in particular vision-language-action (VLA) models and self-supervised predictive architectures (e.g. JEPA), - are able to implement, train, and evaluate learning-based components of a robotic system on real robot arms.	

Learning content	This module covers the following topics: - Foundations of robotics (about 25% of the course): rigid-body motions, forward and inverse kinematics, Jacobians, trajectory generation, and an overview of dynamics and classical control - Deep learning for robot perception: convolutional networks and vision transformers, object detection and segmentation, pose estimation, 3D vision, camera calibration - Imitation learning: behavior cloning, distribution shift, interactive approaches (e.g. DAgger), demonstration collection via teleoperation - Reinforcement learning for robotics: Markov decision processes, value-based methods, policy gradient and actor-critic algorithms, reward design - Simulation and sim-to-real transfer: physics simulators, domain randomization, safety aspects of real-robot training - Foundation models for robotics: vision-language models, vision-language-action (VLA) models, and self-supervised world models such as Joint Embedding Predictive Architectures (JEPA) - Practical and theoretical exercises with two AR4 robot arms (kinematics, calibration, demonstration collection, training and deployment of learned policies)
Requirements for participation	Recommended are knowledge of Python, linear algebra, and elementary probability theory / statistics. Prior knowledge of machine learning is helpful but not required; a self-study crash course on machine learning fundamentals is offered before the start of the lecture.
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 at the beginning of the course.
Useful literature	

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. 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.

Volume Visualization

LV-Nr. 1300132218	**Name** Volume Visualization	**Abbreviation** IVV
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. Data and Computer Science M.Sc. Scientific Computing
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 - Machine Learning for Image Generation	
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	

Modules from the MSc Technische Informatik

Here is only a list of modules. For a detailed description of the modules see the module handbook for the MSc Technische Informatik.

– Introduction to High Performance Computing
– Embedded Machine Learning
– GPU Computing
– Advanced Parallel Computing
– Advanced Parallel Algorithms
– Accelerator Practice
– Parallel Algorithm Design
– Parallel Computer Architecture

3.2 Courses in Mathematics

In this chapter all module descriptions for the modules in mathematics are listed. In each module several courses can be attended.

For the elective compulsory modules in Mathematics 16 CP must be recieved.

The english translation of these descriptions is in progress.

Attention: The course language can be English or German.

Grundmodul Angewandte Analysis und Modellierung

LV-Nr. 11001212XX	**Name** Grundmodul Angewandte Analysis und Modellierung	**Abbreviation** MM12
CP 8 pro Veranstaltung	**Duration** pro Veranstaltung: ein Semester	**Offered** mindestens jährlich
Format pro Veranstaltung: Vorlesung 4 SWS + Übung 2 SWS	**Workload** pro Veranstaltung: 240 h; davon 60 h Präsenz in der Vorlesung 30 h Präsenz in Übungen 120 h Hausaufgaben und selbständiges Nacharbeiten 30 h Prüfungsvorbereitung	**Availability** Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden. M.Sc. Mathematik, M.Sc. Scientific Computing
Language Deutsch oder Englisch	**Lecturer(s)** wechselnd	**Examination scheme** 1+1 pro Veranstaltung
Learning objectives	- Verständnis der grundlegenden Strukturen, Sätze und Methoden eines Forschungsgebietes der Mathematik, - Fähigkeit, typische Aussagen mit den erlernten Methoden selbständig zu beweisen, eigene Kenntnislücken zu erkennen und selbständig zu schließen, - Selbstbewusster Umgang mit Lernstrategien und mathematischem Denken.	

Learning content	In diesem Modul werden folgende Veranstaltungen angeboten: Elliptische partielle Differentialgleichungen: Existenz von Lösungen linearer elliptischer Differentialgleichungen, Höhere Regularität in Sobolevräumen, Cacciopoli-Leray Ungleichung, Schaudertheorie, Campanatoräume, BMO, L^p-Theorie elliptischer Differentialgleichungen, Harmonischen Abbildungen. Evolutionsgleichungen: Bochner Integral, Aubin-Lions Lemma, Galerkinverfahren, Schwache Lösung für Parabolische Differentialgleichungen, Hyperbolische Differentialgleichungen, Navier Stokes Gleichung, Euler-Gleichung, weitere Beispiele nichtlinearer Differentialgleichungen Nichtlineare Funktionalanalysis: Fixpunktsatz von Schauder, Theorie des Abbildungsgrades, Lemma von Sard, Theorie monotoner Operatoren, Anwendungen auf partielle Differentialgleichungen, Bifurkationstheorie, Hopf-Bifurkation Variationsrechnung und Modellierung: Variationsrechnung in mehreren Variablen, Motivierung aus Systemen der Natur, Direkte Methode, Euler-Lagrange Gleichung, Null-Lagrangians, Konvexitätsbegriffe, Gamma-Konvergenz, Homogenisierung, Gradientenflüsse Nonlinear Partial Differential Equations: Investigated PDEs include a subset of prototypical models from different topics such as Gradient Flows, Fluid Dynamics, Reaction-Diffusion equations, Hanilton-Jacobi Equations Partial Differential Equations in Data Science (MLC3) aus dem MMLDS: Introduction to Calculus of Variations, Introduction to Optimal Transport, some measure theoretical methods for PDEs, PDEs on measure spaces
Requirements for participation	empfohlen sind: Kenntnisse der Analysis, linearen Algebra, Numerik und Funktionalanalysis
Requirements for the assignment of credits and final grade	Jede Veranstaltung wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.

Useful literature	wird in HeiCO oder auf der Homepage der Vorlesung angegeben

Basic course Numerical Analysis and Optimization

LV-Nr. 11001215XX	**Name** Basic course Numerical Analysis and Optimization	**Abbreviation** MM15
CP 8 pro Veranstaltung	**Duration** pro Veranstaltung: ein Semester	**Offered** mindestens jährlich
Format pro Veranstaltung: Vorlesung 4 SWS + Übung 2 SWS	**Workload** pro Veranstaltung: 240 h; davon 60 h Präsenz in der Vorlesung 30 h Präsenz in Übungen 120 h Hausaufgaben und selbständiges Nacharbeiten 30 h Prüfungsvorbereitung	**Availability** Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden. M.Sc. Mathematik, M.Sc. Scientific Computing
Language Deutsch oder Englisch	**Lecturer(s)** wechselnd	**Examination scheme** 1+1 pro Veranstaltung
Learning objectives	- Verständnis der grundlegenden Strukturen, Sätze und Methoden eines Forschungsgebietes der Mathematik, - Fähigkeit, typische Aussagen mit den erlernten Methoden selbständig zu beweisen, eigene Kenntnislücken zu erkennen und selbständig zu schließen, - Selbstbewusster Umgang mit Lernstrategien und mathematischem Denken.	

Learning content	In diesem Modul werden folgende Veranstaltungen angeboten: Finite Elemente: Überblick über die Theorie schwacher Lösungen elliptischer Differentialgleichungen, Galerkinapproximation von Variationsproblemen, Aufbau der Methode der finiten Elemente, das Bramble-Hilbert-Lemma, a priori und a posteriori Fehleranalyse, Lösung der diskreten Probleme, Mehrgitterverfahren, Aspekte der Implementation, adaptive Gitterverfeinerung, Einführung in parabolische Gleichungen Nichtlineare Optimierung: Endlich-dimensionale, glatte, kontinuierliche, nichtlineare Optimierungsprobleme, Optimalitäsbedingungen für unbeschränkte und beschränkte Optimierungsprobleme, Gradientenverfahren, Konjugierte Gradienten-(CG-)Verfahren, Line Search, Newton- und Quasi-Newton-SQP-Verfahren, Gauß-Newton-Verfahren, Behandlung von Ungleichungsbeschränkungen, Trust-Region-Verfahren, Automatische Differentiation Numerische Optimierung bei Differentialgleichungen I: Modellierung dynamischer Prozesse, Parameterschätzung (Einfachschießverfahren, Mehrzielmethode, Kollokation, Verallgemeinertes Gauß-Newton, Strukturausnutzende Lösung der linearisierten Subprobleme, Konvergenzeigenschaften), Optimalsteuerungsproblem (Problemformulierung, Direkte Methode zur Lösung von Optimalsteuerungsproblemen, Mehrzielmethode, SQP-Verfahren, Strukturausnutzende SQP-Verfahren für das diskretisierte Optimalsteuerungsproblem) Uncertainty Quantification 1: Im Rahmen dieser Veranstaltung werden methodische Ansätze vermittelt, die es ermöglichen, eine Quantifizierung der Unsicherheit im Zusammenhang mit komplexen numerischen Modellen zu gewinnen. Folgende Schwerpunkte werden behandelt: Rundungsfehler und Fehlerfortpflanzung in der Numerik, Kondition eines Problems; Stabilitätskonzepte, Monte-Carlo Methoden und Kollokationsverfahren, Polynomielle Chaosentwicklungen, Stochastische Galerkin Diskretisierung
Requirements for participation	empfohlen sind: Kenntnisse der Analysis, linearen Algebra und Numerik.

Requirements for the assignment of credits and final grade	Jede Veranstaltung wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.
Useful literature	wird in HeiCO oder auf der Homepage der Vorlesung angegeben

Grundmodul Statistik und Wahrscheinlichkeitsrechnung

LV-Nr. 11001216XX	**Name** Grundmodul Statistik und Wahrscheinlichkeitsrechnung	**Abbreviation** MM16
CP 8 pro Veranstaltung	**Duration** pro Veranstaltung: ein Semester	**Offered** mindestens jährlich
Format Lecture 4 SWS + Tutorial 2 SWS	**Workload** pro Veranstaltung: 240 h; davon 60 h Präsenz in der Vorlesung 30 h Präsenz in Übungen 120 h Hausaufgaben und selbständiges Nacharbeiten 30 h Prüfungsvorbereitung	**Availability** Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden. M.Sc. Mathematik, M.Sc. Scientific Computing
Language Deutsch oder Englisch	**Lecturer(s)** wechselnd	**Examination scheme** 1+1 pro Veranstaltung
Learning objectives	- Verständnis der grundlegenden Strukturen, Sätze und angewandten und theoretischen Methoden der Wahrscheinlichkeitstheorie und/oder Statistik, - Fähigkeit, theoretische Aussagen mit den erlernten Methoden selbständig zu beweisen und die Kenntnisse in praktischen Kontexten anzuwenden.	

Learning content	In diesem Modul werden folgende Veranstaltungen angeboten: Wahrscheinlichkeitstheorie II: Theorie stochastischer Prozesse (Endlich-dimensionale Verteilungen, Existenzsatz von Kolmogorov, stetige Pfade, Konstruktion und Eigenschaften der Brownschen Bewegung, Gaußprozesse); Ergodentheorie (Stationäre und ergodische Prozesse, Ergodensätze); Invarianzprinzipien (Straffheit, schwache Konvergenz im Raum der stetigen Funktionen, Invarianzprinzip von Donsker, Theorie der empirischen Prozesse); stochastisches Integral (Martingale in stetiger Zeit, Itô-Integral, Itô-Formel) Mathematical statistics: decision theory and asymptotics: 1 Part Decision theory 1.1 Decision theory: statistical experiment, Minimax- und Bayes approach, Stein phenomenom 1.2 Estimatiion theory: sufficiency, completeness, exponential families, unbiasedness, information inequalities, translations-equivariance 1.3 Test theory: Neyman-Pearson, conditional tests, likelihood-ratio-tests, rank-tests 2 Part Asymptotic theory 2.1 General estimation methods: moment estimators, MLE, Minimum-kontrast (M and Z- estimators), consistency, aqsymoptotic normality 2.2 Asymptotic properties of tests: Contiguity, Local asymptotic normality (LAN), Asymptotic relative efficiency, Asymptotic power of rank tests, LeCam Theory; Optimality theory in LAN experiments Statistical Learning (MLC4) aus dem MMLDS: I.Statistical learning: Linear models, high-dimensional models, Lasso, kernel methods for nonparametric density and data fitting II. Empirical process theory: Uniform laws of large numbers (Bracketing, finite dimensional approximation), Symmetrisation, Univariate exponential inequalities (Inequalities of Hoeffding, Bennett, Prokhorov, Bernstein), Set indexed empirical processes (Glivenko-Cantelli classes, Vapnik-Cervonenkis classes), Uniform laws of large numbers (Covering numbers), Concentration inequalities (Tsirelson-Ibragimov-Sudakov inequality, Talagrand?s inequality)
Requirements for participation	empfohlen sind Kenntnisse der Analysis und Linearen Algebra, Wahrscheinlichkeitstheorie 1

Requirements for the assignment of credits and final grade	Jede Veranstaltung wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.
Useful literature	wird in HeiCO oder auf der Homepage der Vorlesung angegeben

Aufbaumodul Angewandte Analysis und Modellierung

LV-Nr. 11001222XX	**Name** Aufbaumodul Angewandte Analysis und Modellierung	**Abbreviation** MM22
CP 8 pro Veranstaltung	**Duration** pro Veranstaltung: ein Semester	**Offered** mindestens jährlich
Format pro Veranstaltung: Vorlesung 4 SWS + Übung 2 SWS	**Workload** pro Veranstaltung: 240 h; davon 60 h Präsenz in der Vorlesung 30 h Präsenz in Übungen 120 h Hausaufgaben und selbständiges Nacharbeiten 30 h Prüfungsvorbereitung	**Availability** Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden. M.Sc. Mathematik M.Sc. Scientific Computing
Language Deutsch oder Englisch	**Lecturer(s)** wechselnd	**Examination scheme** 1+1 pro Veranstaltung
Learning objectives	- Vertieftes Verständnis der Strukturen, Sätze, Beweise und Methoden eines engeren Forschungsgebietes der Mathematik, - Fähigkeit, Aussagen aus dem Teilgebiet selbständig zu beweisen und Beweistechniken zu diskutieren, sowie Aufgaben auf ihre Charakteristika hin zu analysieren und zu klassifizieren um geeignete Lösungsmethoden zu wählen, - Fähigkeit, sich Teilaspekte des Themengebietes selbständig zu erarbeiten.	
Learning content	In diesem Modul werden folgende Veranstaltungen angeboten: Mathematische Grundlagen der Fluid Dynamik: Physikalische Motivation der Navier-Stokes Gleichung, Spezielle Lösungen, Kurzzeitexistenz schwacher Lösung, Langzeitexistenz schwacher Lösungen, Vortizität, Navier-Stokes Gleichung in zwei Dimensionen, Existenz von Lösungen der Eulergleichung PDGs und Modellierung: Modellierung physikalischer/biologischer Prozesse (z.B. Fluiddynamik, Materialwissenschaften, Biologie, ...), Grundlegende mathematische Theorie	
Requirements for participation	Grundmodul Angewandte Analysis und Modellierung	

Requirements for the assignment of credits and final grade	Jede Veranstaltung wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.
Useful literature	wird in HeiCO oder auf der Homepage der Vorlesung angegeben

Advanced course numerical analysis and optimization

LV-Nr. 11001225XX	**Name** Advanced course numerical analysis and optimization	**Abbreviation** MM25
CP 8 pro Veranstaltung	**Duration** pro Veranstalung: ein Semester	**Offered** mindestens jährlich
Format pro Veranstaltung: Vorlesung 4 SWS + Übung 2 SWS	**Workload** pro Veranstaltung: 240 h; davon 60 h Präsenz in der Vorlesung 30 h Präsenz in Übungen 120 h Hausaufgaben und selbständiges Nacharbeiten 30 h Prüfungsvorbereitung	**Availability** Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden. M.Sc. Mathematik M.Sc. Scientific Computing
Language Deutsch oder Englisch	**Lecturer(s)** wechselnd	**Examination scheme** 1+1 pro Veranstaltung
Learning objectives	- Vertieftes Verständnis der Strukturen, Sätze, Beweise und Methoden eines engeren Forschungsgebietes der Mathematik, - Fähigkeit, Aussagen aus dem Teilgebiet selbständig zu beweisen und Beweistechniken zu diskutieren, sowie Aufgaben auf ihre Charakteristika hin zu analysieren und zu klassifizieren um geeignete Lösungsmethoden zu wählen, - Fähigkeit, sich Teilaspekte des Themengebiets selbständig zu erarbeiten.	

Learning content	In diesem Modul werden folgende Veranstaltungen angeboten: - Gemischte Finite Elemente: Stokes- und Navier-Stokes-Gleichungen, Sattelpunktprobleme, das closed range theorem und inf-sub-Stabilität, Taylor-Hood- Elemente, Darcy- Gleichungen für Strömung durch poröse Medien, finite element exterior calculus, discontinuous Galerkin methods - Parallele Löser für Finite Elemente: abstrakte Unterraumkorrekturverfahren, überlappende Schwarz-Verfahren, geometrische und algebraische Mehrgitter- verfahren, nichtüberlappende Gebietszerlegungsverfahren, Konvergenztheorie der Unterraum-korrekturverfahren, Implementation und parallele Skalierbarkeit - Numerische Optimierung bei Differentialgleichungen II: Parameterschätzung mit Beschränkungen und Konvergenzanalyse für Verallgemeinerte (beschränkte) Gauß-Newton-Verfahren, Statistische Sensitivitätsanalyse (Vertrauens-/ Konfidenzgebiete, Kovarianz-Analyse), optimale Versuchsplanung (Problemformulierung, Sequentielle Versuchs-planung, Numerische Lösung mit SQP-Verfahren, effiziente Ableitungs-berechnung), Globalisierung der Konvergenz bei Newton-Verfahren für sehr nichtlineare Probleme (Abstiegsstrategien, Natürliche Niveaufunk- tionen, Restriktiver Monotonie-Test (RMT) und praktische Realisierung), Fortsetzungsmethoden (Allgemeine Strategie, Verfahren höherer Ordnung, Schrittweitensteuerung), Effiziente Ableitungsberechnung (Vorwärts- und Rückwärtsmodus, Anwendung auf gewöhnliche Differentialgleichungen und Diskretisierungs-Verfahren dafür) - Unendlich-dimensionale Optimierung: Existenz optimaler Lösungen, Differenzierbarkeitsbegriffe, notwendige und hinreichende Optimalitätsbedingungen, numerische Lösungsverfahren, Anwendungen in der optimalen Steuerung und Bildverarbeitung - Informationsgeometrie und Maschinelles Lernen: Differentialgeometrie (Mannigfaltigkeiten, Untermannigfaltigkeiten, Vektor-, Kovektor- und Tensorfelder, Riemannsche Metriken, affine Zusammenhänge, Geodäten, Krümmungstensor), Informationsgeometrie (Maße auf endlichen Mengen, Fisher-Rao Metrik, alpha-Zusammenhänge, Divergenzfunktionen, Informationsprojektionen, graphische Modelle, Exponentialfamilie, statistische Mannigfaltigkeiten), Maschinelles Lernen (ausgewählte Probleme der Inferenz, des überwachten und unbewachten Lernens als Anwendungsbeispiele) - High-dimensional Numerics (MLC2) aus dem MMLDS: Uncertainty Quantification (Parametric Models, Model problems, Parameter regularity), High-dimensional integration (Monte Carlo methods, Sampling from random fields, Quasi-Monte Carlo, Sparse-grids, Multilevel methods), High-dimensional approximation (Regularity for infinite-parametric functions, Stochastic collocation, Low-rank tensor approximation, Random features) - Variational Methods and Optimization in Machine Learning (MLC5) aus dem MMLDS: The lecture covers variational and optimization methods in the context of major tasks of data science and machine learning: data embedding and representation; supervised, semi-supervised and unsupervised learning; classification, regression and density estimation using discriminative and generative models; structured prediction. Besides established classical methods, deep learning will be a central theme, including applications to numerical optimization.

Requirements for participation	empfohlen ist: Grundmodul Numerik und Optimierung
Requirements for the assignment of credits and final grade	Jede Veranstaltung wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.
Useful literature	wird in HeiCO oder auf der Homepage der Vorlesung angegeben

Aufbaumodul Statistik und Wahrscheinlichkeitsrechnung

LV-Nr. 11001226XX	**Name** Aufbaumodul Statistik und Wahrscheinlichkeitsrechnung	**Abbreviation** MM26
CP 8 pro Veranstaltung	**Duration** pro Veranstaltung: ein Semester	**Offered** mindestens jährlich
Format pro Veranstaltung: Vorlesung 4 SWS + Übung 2 SWS	**Workload** pro Veranstaltung: 240 h; davon 60 h Präsenz in der Vorlesung 30 h Präsenz in Übungen 120 h Hausaufgaben und selbständiges Nacharbeiten 30 h Prüfungsvorbereitung	**Availability** Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden. M.Sc. Mathematik M.Sc. Scientific Computing
Language Deutsch oder Englisch	**Lecturer(s)** wechselnd	**Examination scheme** 1+1 pro Veranstaltung
Learning objectives	- Vertieftes Verständnis der grundlegenden Strukturen, Sätze und angewandten und theoretischen Methoden der Wahrscheinlichkeitstheorie und/oder Statistik, - Fähigkeit, theoretisch zu argumentieren, neue Aussagen mit den erlernten Methoden selbständig zu beweisen und das Potential der Methoden in praktischen Kontexten zu erkennen.	

Learning content	In diesem Modul werden folgende Veranstaltungen angeboten: - Fortgeschrittene Zeitreihenanalyse - Statistik zeitstetiger Prozesse - Angewandte Statistik - Lokale asymptotische Normalität und Semiparametrik: Asymptotische Entscheidungstheorie für lokal asymptotisch normale Experimente, Differenzierbarkeit im quadratischen Mittel, Kontiguität, Semiparametrik, asymptotische Effizienz in semiparametrischen Modellen - Empirische Prozesse: Glivenko-Cantelli Sätze, Vapnik-Cervonenskis Theorie, Konzentrationsungleichungen für empirische Prozesse, Donsker Theoreme, Entropieabschätzungen für Funktionenklassen, Konvergenzraten in der Nichtparametrik - Nichtparametrische Minimaxtheorie - Statistik inverser Probleme: Lineare schlecht-gestellte inverse Probleme, spektrale Regularisierungsverfahren, Projektionsverfahren, linearer Galerkinansatz, nicht-parametrische Kurvenschätzung, Orakel-Optimalität, Minimax Theorie, Datengetriebene Schätzverfahren, Gauß'sche inverse Regression, Dekonvolution, funktionale lineare Regression, nicht-parametrische instrumentale Regression - Bayesstatistik - Hoch-dimensionale Statistik: Hoch-dimensionale lineare Modelle, Schätzverfahren in hoch-dimensionalen linearen Modellen, insbesonders LASSO-Schätzer, Konfidenzbereiche und Testverfahren in hoch-dimensionalen linearen Modellen, Modellwahlverfahren, Kleinste Quadrate Schätzer mit Komplexitätsstraftermen, Klassifikationsverfahren
Requirements for participation	empfohlen ist eine Veranstaltung des Grundmoduls Statistik und Wahrscheinlichkeitsrechnung
Requirements for the assignment of credits and final grade	Jede Veranstaltung wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.
Useful literature	wird in HeiCO oder auf der Homepage der Vorlesung angegeben

Spezialisierungsmodul Angewandte Analysis und Modellierung

LV-Nr. 11001232XX	**Name** Spezialisierungsmodul Angewandte Analysis und Modellierung	**Abbreviation** MM32
CP	**Duration** pro Veranstaltung: ein Semester	**Offered** mindestens jährlich
Format	**Workload**	**Availability** Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden. M.Sc. Mathematik M.Sc. Scientific Computing
Language Deutsch oder Englisch	**Lecturer(s)** wechselnd	**Examination scheme** 1+1 pro Veranstaltung
Learning objectives	- Umfassende Kenntnisse und Verständnis der Strukturen, Aussagen, Methoden und Beweistechniken eines aktuellen Forschungsthemas der Mathematik, - Fähigkeit, sich komplexe mathematische Sachverhalte selbst zu erarbeiten und zu diskutieren.	
Learning content	aktuelle Forschungsthemen aus den Arbeitsgebieten der Dozierenden	
Requirements for participation	empfohlen sind Veranstaltung(en) aus dem Aufbaumodul Analysis und Modellierung	
Requirements for the assignment of credits and final grade	Jede Veranstaltung wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.	
Useful literature	wird in HeiCO oder auf der Homepage der Vorlesung angegeben	

Special topics of numerical analysis and optimization

LV-Nr. 11001235XX	**Name** Special topics of numerical analysis and optimization	**Abbreviation** MM35
CP	**Duration** pro Veranstaltung: ein Semester	**Offered** mindestens jährlich
Format	**Workload**	**Availability** Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden. M.Sc. Mathematik M.Sc. Scientific Computing
Language Deutsch oder Englisch	**Lecturer(s)** wechselnd	**Examination scheme** 1+1 pro Veranstaltung
Learning objectives	- Umfassende Kenntnisse und Verständnis der Strukturen, Aussagen, Methoden und Beweistechniken eines aktuellen Forschungsthemas der Mathematik, - Fähigkeit, sich komplexe mathematische Sachverhalte selbst zu erarbeiten und zu diskutieren.	
Learning content	Aktuelle Forschungsthemen aus den Arbeitsgebieten der Dozierenden. Angeboten werden folgende Veranstaltungen: Fundamentals of Computational Environmental Physics (every wintersemester, 4 SWS lecture + 2 SWS exercise session, 8 LP): Elementary linear models: Flow in porous media, elliptic partial differential equations (PDEs), Scalar transport, first-order hyperbolic PDEs, Contaminant Transport, parabolic PDEs, Coupled elementary models, active transport, Fluid dynamics, Stokes and Navier-Stokes equations	
Requirements for participation	empfohlen sind Veranstaltung(en) aus dem Aufbaumodul Numerik und Optimierung	
Requirements for the assignment of credits and final grade	Jede Veranstaltung wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.	
Useful literature	wird in HeiCO oder auf der Homepage der Vorlesung angegeben	

Spezialisierungsmodul Statistik und Wahrscheinlichkeitsrechnung

LV-Nr. 11001236XX	**Name** Spezialisierungsmodul Statistik und Wahrscheinlichkeitsrechnung	**Abbreviation** MM36
CP	**Duration** pro Veranstaltung: ein Semester	**Offered** mindestens jährlich
Format	**Workload**	**Availability** Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden. M.Sc. Mathematik M.Sc. Scientific Computing
Language Deutsch oder Englisch	**Lecturer(s)** wechselnd	**Examination scheme** 1+1 pro Veranstaltung
Learning objectives	- Umfassende Kenntnisse und Verständnis der Strukturen, Aussagen, Methoden und Beweistechniken der Statistik und Wahrscheinlichkeitstheorie, - Fähigkeit, sich komplexe mathematische Sachverhalte selbst zu erarbeiten und zu diskutieren.	
Learning content	aktuelle Forschungsthemen aus den Arbeitsgebieten der Dozierenden	
Requirements for participation	empfohlen sind Veranstaltung(en) aus dem Aufbaumodul Statistik und Wahrscheinlichkeitsrechnung	
Requirements for the assignment of credits and final grade	Jede Veranstaltung wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Weitere Details werden von der bzw. dem Lehrenden zu Beginn der Veranstaltung bekannt gegeben.	
Useful literature	wird in HeiCO oder auf der Homepage der Vorlesung angegeben	

Applied Mathematics in Biology: Single-cell omics

LV-Nr. L 1100124000 E 1100124001	**Name** Applied Mathematics in Biology: Single-cell omics	**Abbreviation**
CP 6	**Duration**	**Offered** every summersemester
Format Lecture 2 SWS + Exercise 2 SWS	**Workload** 180 h; thereof 30 h lecture 30 h exercise 15 h preparation for exam 105 h self-study and working on assignments (optionally in groups)	**Availability** M.Sc. Mathematik M.Sc. Scientific Computing
Language English	**Lecturer(s)** Simon Anders	**Examination scheme** 1+1
Learning objectives	Modern molecular biology has much need for computational expertise, providing opportunities for researchers from mathematics, physics, scientific computing or other computational sciences. This lecture aims to provide knowledge and skills to get started with such interdisciplinary work – by diving into one specific topic, namely the analysis of single-cell omics data, i.e., mRNA sequencing data with single-cell resolution, a technique used to characterize the molecular state of cells in tissue samples, in order to understand biological processes in health and disease. Students will learn how to work in a foreign scientific domain of knowledge (biology), apply mathematical concepts there, bridge terminology gaps and other overcome challenges of interdisciplinary work. We will explore "'real-world"' applications of mathematical methods, especially from high-dimensional statistics, and learn how to use exploratory data analysis to turn domain-language (here: biological) questions into mathematical hypotheses on the structure of the given data manifolds, and how to translate results back to make them understandable for domain experts. Students will gain practical experience in scientific programming, applied linear algebra, machine learning, data reduction and interpretation, and, most importantly, in interdisciplinary research work, as well as insights into opportunities for computational scientists to work in the life sciences.	

Learning content	Content of this lecture is: - basic concepts of molecular biology and transcriptomics - assay techniques in single cell (sc) biology - stochastic models for omics data - the concept of feature space, as used for omics data and in machine learning - methods to explore high-dimensional data - linear and non-linear methods for dimension reductions - clustering in high-dimensional space - non-linear regression methods - graph-based methods in omics data analysis - applications of machine learning methods to sc omics - deep learning (esp. variational autoencoders) and related methods - techniques for handling big data - interactive visualization for exploratory data analysis
Requirements for participation	recommended is a basic knowledge in (finite-dimensional) linear algebra (matrix calculus, eigendecomposition, etc), experience with at least one programming language and interest in biology
Requirements for the assignment of credits and final grade	The module is completed with a graded written or oral examination. The grade of this examination gives the grade for this module. Details for this examination as well as the requirements for the assignment of credits will be given by the lecturer an the beginning of this course.
Useful literature	

4 General Modules

This chapter concludes the descriptions of the mandatory modules *Master Seminar (Mathematics)*, *Master Programming Practical*, *Master Thesis* and *Master Thesis Presentation*.
Another mandatory module is the *Field of Application* which allows individual choices. A possibility to receive the CP in the Field of Application is the new module *Lab internship in field of application*.

Master Seminar (Mathematics)

LV-Nr.	**Name** Master Seminar (Mathematics)	**Abbreviation**
CP 6	**Duration**	**Offered** every semester
Format Seminar 2 SWS + tutorial 2 SWS	**Workload** 180h; thereof 60h seminar and tutorial 120h preparation of presentation and supervision	**Availability** M.Sc. Scientific Computing
Language English or German	**Lecturer(s)** depending on teaching offer	**Examination scheme**
Learning objectives	- Ability to read mathematical literature (usually a more demanding text), to deal with a mathematical problem independently and to present it, - Ability to communicate mathematical arguments clearly and understandably to a smaller circle of peers, - Knowledge of techniques of scientific writing (especially literature research) Ability to work on complex scientific literature, - Advanced ability to present complex scientific literature in a lecture, - Advanced ability to discuss lectures and give feedback, - Ability to create a short scientific paper on a complex topic.	
Learning content	- After consultation with the lecturer, in particular a comprehensive counseling interview preceding the presentation, - Introduction to and practice of scientific writing techniques, - In-depth training in the development and presentation of scientific literature.	
Requirements for participation	will be announced by the lecturer	
Requirements for the assignment of credits and final grade	The module is completed with a graded exam. This exam includes the presentation of an about 45- to 90-minute presentation, active and passive participation in other lectures and a written elaboration of the presentation (about 10 pages). The appropriate scope, form and content are assessed and graded. The exam must be passed in order to be awarded the LP. The final grade of the module is determined by the grade of the exam.	
Useful literature		

Master Programming Practical

LV-Nr.	**Name** Master Programming Practical	**Abbreviation**
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. Scientific Computing
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	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 for this module is the grade of this examination. More details will be given by the lecturer.	

Useful literature	

Master Thesis

LV-Nr.	**Name** Master Thesis	**Abbreviation**
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. Scientific Computing
Language German or English	**Lecturer(s)** varying	**Examination scheme** 1+1
Learning objectives	- Ability to carry out research work in a specific research area of Scientific Computing under supervision, - Ability to independently produce a scientific thesis.	
Learning content	research work on a specific topic in Scientific Computing	
Requirements for participation	45 credits, according to the examination regulations §17 (2)	
Requirements for the assignment of credits and final grade	Passing the graded Master thesis is required for the award of the CP. This includes regular consulting with advisor and the written elaboration.	
Useful literature	suggested by the advisor	

Master Thesis Presentation

LV-Nr.	Name Master Thesis Presentation	Abbreviation
CP 6	Duration	Offered continuous
Format Seminar	Workload 180h; Preparation of the presentation, delivery and subsequent discussion	Availability M.Sc. Scientific Computing
Language (usually in) English	Lecturer(s) thesis advisor	Examination scheme 1+1
Learning objectives	The students - acquire, practice and demonstrate the ability to present their own research in a scientific presentation, - are able to put their own work in the context of the current state of the research field and to communicate this, - gain skills and experience in disussing questions regarding their own research work.	
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 field, - Discussion of questions of different levels by peers and academics.	
Requirements for participation	completed master's thesis (recommended), see §19 (1) examination regulations	
Requirements for the assignment of credits and final grade	The module is completed with a graded evaluation of the presentation (approximately 30-60 minutes) and the student's ability to defend the results of his/her work in the face of questions and comments (approximately 15-45 minutes). Total time should not exceed 90 minutes. The grade must be at least a pass for the CP to be awarded. The final grade of the module is determined by the grade of the evaluation.	
Useful literature		

Field of Application

LV-Nr.	Name Field of Application	Abbreviation
CP 16	Duration	Offered
Format	Workload 480 h; Division into presence, practice time, internship, exercises and consultation with the lecturer/supervisor.	Availability M.Sc. Scientific Computing
Language English or German	Lecturer(s)	Examination scheme
Learning objectives	in-depth knowledge and skills in an application area	
Learning content	Selection of one field of applications to demonstrate applicability of scientific computing tools and techniques. Selectable are general fields of research, both disciplinary and interdisciplinary: Physics, chemistry, bio-sciences, astronomy, economics, computational linguistic, medical physics are examples of fields of application. Modules from the field of application have to be from a master program of Heidelberg University. Bachelor modules can be allowed by the dean of studies if they are needed as prerequisites to attend master modules. Modules have to have their focus on the filed of application - modules teaching mainly mathematics and/or computer science cannot be credited.	
Requirements for participation		
Requirements for the assignment of credits and final grade	according to the regulations of the respective department	
Useful literature		

Lab internship in field of application

LV-Nr.	**Name** Lab internship in field of application	**Abbreviation**
CP 10	**Duration** 10 weeks	**Offered**
Format Internship	**Workload** 300 h; thereof 240 h lab internship according to a lab internship proposal. 60 h documentation and presentation of proposal results	**Availability** M.Sc. Scientific Computing
Language English or German	**Lecturer(s)**	**Examination scheme**
Learning objectives	Students defines a research projects, does independent research under supervision, documents results in written form and presents results to peer group.	
Learning content	Student gets a research project at a work group in the field of application. The internship is supervised by a lecturer, postdoc or PhD student under the guidance of the group leader. The research project is carried out in 8 weeks under supervision as member of the research group. Student is fully immersed as part of the research group and participates in daily group life. The content of the lab internship is defined in a lab project proposal which is defined and signed by student and head of the lab and confirmed by the dean of studies.	
Requirements for participation	Preliminaries to complete the defined research project	
Requirements for the assignment of credits and final grade	The module is completed with a graded exam. This exam includes the completed project according to proposal, the presentation and the project documentation.	
Useful literature		

5 Key Competence Program

Interdisciplinary competencies (in German *Übergreifende Kompetenzen ÜK*) refer to study contents, key competences 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 competences (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 scientific computing 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. One year of organizational work in a university body, e.g. in the SIAM student chapter, can also be recognized, it is essential to consult with the Examination Office. Furthermore, irregular offers of the faculty marked as ÜK can be taken.
From the Master of Computer Engineering the modules *Entrepreneurship*, *Tools* and *C++ Practice* can be chosen. For the module description please refer to the module handbook of the Master Computer Engineering course of studies.

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. 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		

Industrial Internship for Scientific Computing

LV-Nr.	**Name** Industrial Internship for Scientific Computing	**Abbreviation**
CP	**Duration**	**Offered**
Format Working in an industrial company	**Workload** at least 120 hours, of which at most 10% on the report.	**Availability** M.Sc. Scientific Computing
Language	**Lecturer(s)** Chairperson of the examination board	**Examination scheme** 1+1
Learning objectives	Learning and application of mathematical and/or computational methods in an industrial context.	
Learning content	The industrial internship is intended to impart project-related application of scientific computing. Ideally, the internship should be embedded in a process (e.g., in implementation of algorithms) 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	The awarding of the CP does not only depend on the duration (time commitment) of the internship, but also on the content. 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	For this purpose, an approx. 6-page, well-structured written report on the activities carried out, including the task 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. 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		

Introduction to LaTeX

LV-Nr. L 1100180005 E 1100180006	**Name** Introduction to LaTeX	**Abbreviation** MLat
CP 3 ÜK	**Duration** one semester	**Offered** annually
Format Lecture 2 SWS + Exercise 2 SWS	**Workload** 90 h; thereof 30 h lecture 30 h exercise 30 h self-study	**Availability** B.Sc. Informatik B.Sc. Mathematik M.Sc. Data and Computer Science M.Sc. Scientific Computing M.Sc. Mathematics of Machine Learning and Data Science
Language English	**Lecturer(s)** varying	**Examination scheme**
Learning objectives	Students are able to - configure a LaTeX system (locally or online) - structure and edit complex LaTeX projects - identify and correct compile errors - select and utilize external packages - implement and use LaTeX macros	
Learning content	The course offers an introduction into the typesetting system LaTeX with a focus on its effective use in an academic and professional context, based on either a local installation or an online platform. Contents include - getting started with LaTeX - document structure and formatting - floating objects (figures, tables) - handling of bibliographies and citations - typesetting mathematical formulas - incorporation of external packages - graphical illustrations using TikZ - important types of documents (e.g., presentations, theses, CVs, letters) - development of custom macros and packages - directories and search paths	
Requirements for participation	none	
Requirements for the assignment of credits and final grade	Active participation in lectures and exercises	

Useful literature	to be announced in the lecture

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. 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), latest edition	

Software Economics

LV-Nr. 1100222026	**Name** Software Economics	**Abbreviation** 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. 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 (ISE)
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