Module Handbook

International Master "Scientific Computing"

Ruprecht-Karls-Universität Heidelberg
Fakultät für Mathematik und Informatik

Version as of 09.02.2022 corresponding to examination regulations of 22.04.2013

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

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 competencies 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, deviations from the modalities for exam 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 here in the module descriptions.

Rules for awarding the CP: In this module, the CP are awarded if the final exam is passed. The details of the final exam are described in the individual module descriptions. In this module there is a practice operation with the processing of exercises. In order to be admitted to the final exam, generally 50% of the points in the exercises should be achieved. The lecturer can deviate from this in individual cases.

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

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

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

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

2 Model study plan and Mobility

2.1 Model study plan

1st year:		
	Compulsory optional modules in Mathematics	16 CP
	Compulsory optional modules in Computer Science	16 CP
	Field of Application	18 CP
	2 Seminars	12 CP
sum		**62 CP**
2nd year:		
	Specialization modules	16 CP
	Key Competence Program	6 CP
	Seminar	6 CP
	Master Thesis	30 CP
sum		**58 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 Core 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 compulsory optional modules in Computer Science 16 CP must be recieved.

Advanced Machine Learning

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

Algorithm Engineering

<table>
<tr><td>Code
IAE</td><td colspan="2">Name
Algorithm Engineering</td></tr>
<tr><td>CP
8</td><td>Duration
one semester</td><td>Offered
every summer semester</td></tr>
<tr><td>Format
Lecture 4 SWS
+ Exercise
course 2 SWS</td><td>Workload
240h; thereof
90h lectures and tutorials,
15h exam preparations,
135h lecture wrap-up and homework</td><td>Availability
M.Sc. Angewandte Informatik
M.Sc. Data and Computer Science
M.Sc. Scientific Computing</td></tr>
<tr><td>Language
English</td><td>Lecturer(s)
Christian Schulz</td><td>Examination scheme
1+1</td></tr>
<tr><td>Learning objectives</td><td colspan="2">Students obtain a systematic understanding of algorithmic questions and solution approaches in the area of algorithm engineering.
The students will be able to transfer the learned techniques onto similar problems and be able to interpret and understand current research topics in the area of algorithm engineering.
Given a real-world problem, students are able to select appropriate algorithms to come up with and implement efficient solutions.
In particular, students know realistic machine models and applications, algorithm design, implementation techniques, experimental methodology and can interpret of measurements.</td></tr>
<tr><td>Learning content</td><td colspan="2">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).</td></tr>
<tr><td>Requirements for participation</td><td colspan="2">recommended are:
Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Mathematik für Informatiker 1 oder Lineare Algebra 1 (MA4), Algorithms and Data Structures 2</td></tr>
<tr><td>Requirements for the assignment of credits and final grade</td><td colspan="2">The module is completed with a graded oral exam. The final grade of the module is determined by the grade of the exam. The requirements for the assignment of credits follows the regulations in section modalities for exams.</td></tr>
</table>

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

Algorithms and Data Structures 2

Code IADS2	**Name** Algorithms and Data Structures 2	
CP 8	**Duration** one semester	**Offered** every winter 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** B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Angewandte Informatik 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 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, - are able to analyse new algorithms as well as analysing their running time, and select appropriate algorithms for applications - are able to apply algorithms and data structures to real-world problems, and can objectively assess the quality of the results	
Learning content	Introduction to Algorithm Engineering: - advanced data structures (efficient addressable priority queues, monotone priority queues, external priority queues), - advances graph algorithms (strongly connected components, shortest paths, maximum flows / min s-t cuts, min-cost flows), techniques to solve problems to optimality (branch-and-bound, branch-and-reduce, dynamic programming, integer linear programming as a modelling tool), - introduction to randomized algorithms, greedy algorithms, approximation algorithms, advanced string algorithms, geometric algorithms, external memory algorithms	
Requirements for participation	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Mathematik für Informatiker 1 (IMI1) oder Lineare Algebra 1 (MA4)	
Requirements for the assignment of credits and final grade	The module is completed with a graded oral exam. The final grade of the module is determined by the grade of the exam. The requirements for the assignment of credits follows the regulations in section modalities for exams.	

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

Complex Network Analysis

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

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

Computer Graphics

Code ICG	Name Computer Graphics	
CP 8	**Duration** one semester	**Offered** every 3rd semester
Format Lecture 4 SWS + Exercise course 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** cannot be combined with Computergraphik 1 and 2 (ICG1, ICG2) B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. Scientific Computing
Language English	**Lecturer(s)** Filip Sadlo	**Examination scheme** 1+1
Learning objectives	The students understand fundamental and advanced concepts of computer graphics. They understand the mathematical fundamentals, data structures, and implementation aspects. They get to know raster graphics, geometric transforms, color perception and color models, and basics of geometric modeling. The students are able to apply these concepts to real-world problems using existing software packages, and develop small programs using OpenGL 4.	
Learning content	- Introduction - Perception and Color - Raytracing - Transformations - Rasterization - OpenGL - Textures - Spatial Data Structures	
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 written or oral exam. The final grade of the module is determined by the grade of the exam. The requirements for the assignment of credits follows the regulations in section modalities for exams.	
Useful literature	P. Shirley, S. Marschner: Fundamentals of Computer Graphics, 3rd Edition, AK Peters OpenGL Specifications(GL 4.5 + GLSL 4.50) http://www.opengl.org/registry/ Optional A. S. Glassner: An Introduction to Ray Tracing, Academic Press T. Akenine-Möller, E. Haines: Real-Time Rendering, AK Peters, 2008	

Fundamentals of Machine Learning

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

Geometric Modeling and Animation

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

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

Hardware Aware Scientific Computing

<table>
<tr><td>Code
IHASC</td><td colspan="2">Name
Hardware Aware Scientific Computing</td></tr>
<tr><td>CP
8</td><td>Duration
one semester</td><td>Offered
irregular</td></tr>
<tr><td>Format
Lecture 4 SWS
+ Exercise
Course 2 SWS</td><td>Workload
240h;thereof
90h lecture
15h preparation for exam
135h self-study and working on
assignments/projects (optionally in groups)</td><td>Availability
M.Sc. Angewandte Informatik
M.Sc. Data and Computer
Science
M.Sc. Scientific Computing</td></tr>
<tr><td>Language
English</td><td>Lecturer(s)
Peter Bastian</td><td>Examination scheme
1+1</td></tr>
<tr><td>Learning objectives</td><td colspan="2">Students are familiar with different forms of parallelism in modern computer architectures. The can exploit this parallelism selecting an appropriate programming model. They are familiar with modelling of parallelism and know fundamental parallel algorithms from scientific computing.</td></tr>
<tr><td>Learning content</td><td colspan="2">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</td></tr>
<tr><td>Requirements for participation</td><td colspan="2">basic knowledge in computer architecture and numerical methods; good programming skills in C++</td></tr>
</table>

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

Machine Learning

<table>
<tr><td>Code
IML</td><td colspan="2">Name
Machine Learning</td></tr>
<tr><td>CP
8</td><td>Duration
one semester</td><td>Offered
in (irregular) alternation with *Fundamentals of Machine Learning* + *Advanced Machine Learning*</td></tr>
<tr><td>Format
Lecture 4 SWS + Exercise course 2 SWS</td><td>Workload
Arbeitsaufwand: 240h, thereof
60h lecture
90h tutorials, homework, lecture wrap-up
90h graded final report</td><td>Availability
cannot be combined with *Fundamentals of Machine Learning* or *Advanced Machine Learning*
M.Sc. Angewandte Informatik
M.Sc. Data and Computer Science
M.Sc. Scientific Computing</td></tr>
<tr><td>Language
English</td><td>Lecturer(s)
Ullrich Köthe</td><td>Examination scheme
1+1</td></tr>
<tr><td>Learning objectives</td><td colspan="2">Students understand a broad range of machine learning concepts, get to know established and advanced learning methods and algorithms, are able to apply them to real-world problems, and can objectively assess the quality of the results.
In addition, students learn how to use Python-based machine learning software such as scikit-learn.</td></tr>
<tr><td>Learning content</td><td colspan="2">This lecture is a compact version of the two-semester course *Fundamentals of Machine Learning* + *Advanced Machine Learning*:
Classification (linear and quadratic discriminant analysis, neural networks, linear and kernelized support vector machines, decision trees and random forests), least squares and regularized regression, Gaussian processes, unsupervised learning (density estimation, cluster analysis, Gaussian mixture models and expectation maximization, principal component analysis, bilinear decompositions), directed probabilistic graphical models, optimization for machine learning, structured learning</td></tr>
<tr><td>Requirements for participation</td><td colspan="2">recommended are: solid knowledge of basic calculus, statistics, and linear algebra</td></tr>
<tr><td>Requirements for the assignment of credits and final grade</td><td colspan="2">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.</td></tr>
</table>

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

Mining Massive Datasets

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

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

Object-Oriented Programming for Scientific Computing

Code IOPSC	**Name** Object-Oriented Programming for Scientific Computing	
CP 6	**Duration** one semester	**Offered** every summer semester
Format Lecture 2 SWS + Exercise on computer 2 SWS	**Workload** 180 h; thereof 60 h lecture 105 h self-study and working on assignments 15 h preparation for exam	**Availability** B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. Scientific Computing
Language English	**Lecturer(s)** Ole Klein	**Examination scheme** 1+1
Learning objectives	The students are proficient in the programming language C++, can assess the performance of different programming techniques, know template programming techniques, and can use the Standard Template Library (STL). They can apply their new skills to solve selected problems of Scientific Computing.	
Learning content	This module deepens the skills in object-oriented programming obtained in the basic lecture Einführung in die Praktische Informatik (IPI) with special emphasis on Scientific Computing: Class concept Dynamic memory allocation Exception handling Resource allocation and initialization Constness Static versus dynamic polymorphism Traits and Policies Standard Template Library Template Metaprogramming Parallel programming techniques	
Requirements for participation	recommended are: Einführung in die Praktische Informatik (IPI), alternatively basic knowledge of an object-oriented programming language	
Requirements for the assignment of credits and final grade	The module is completed with a graded written exam. The final grade of the module is determined by the grade of the exam. The lecture will give the requirements for the assignment of credits.	
Useful literature		

Optimization for Machine Learning

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

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

Scientific Visualization

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

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

Text Analytics

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

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

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
– GPU Computing
– Advanced Parallel Computing
– Accelerator Practice
– Robotics 1 - Kinematics, Dynamics and Motion Planning
– Robotics 2 - Simulation and Optimization in Robotics
– Biomechanics 1
– Robotics Practical for Computer Engineering Master

4 Other Courses in Computer Science

The modules listed in this chapter can also be chosen for computer science.

Attention: The course language is most likely german.

Computational Geometry

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

Computerspiele

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

Praktische Geometrie

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

Randomisierte Algorithmen

Code	**Name**	
IRA	Randomisierte Algorithmen	
CP 6	**Duration** ein Semester	**Offered** mindst. jedes 4. Semester
Format Vorlesung 3 SW + Übung 1 SWS	**Workload** 180 h; davon 60 h Präsenzstudium 40 h Prüfungsvorbereitung 80 h Selbststudium und Bearbeitung der Übungsaufgaben (eventuell in Gruppen)	**Availability** B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. Scientific Computing
Language Deutsch	**Lecturer(s)** Wolfgang Merkle	**Examination scheme** 1+1
Learning objectives	Auf der Grundlage der behandelten Anwendungsbeispiele aus verschiedenen Teilgebieten der Informatik können die Studierenden die probabilistische Betrachtungs- und Vorgehensweise anwenden bei der Konstruktion und Analyse von probabilistischen und deterministischen Algorithmen, auf kombinatorische Fragestellungen, um spieltheoretische Situationen zu analysieren, auf kryptographische Fragestellungen.	
Learning content	Elementare Wahrscheinlichkeitsrechnung Das Tenure-Spiel Derandomisierungstechniken Die probabilistische Methode Byzantinische Übereinkunft Stabile Heiraten und der Gale-Shapley-Algorithmus Das Minimax-Prinzip von Yao Komplexitätsanalyse des randomisierten Sortierens Randomisierte Fehlersuche und -korrektur Das Local-Lemma von Lovasz PAC-Lernen und VC-Dimension Wahrscheinlichkeitsverstärkung und Fehlerschranken Lokale Suche für k-SAT Kryptographische Protokolle	
Requirements for participation	empfohlen sind: elementare Grundkenntnisse in Algorithmen wie sie z.B. im Modul Algorithmen und Datenstrukturen (IAD) vermittelt werden.	
Requirements for the assignment of credits and final grade	Das Modul wird mit einer benoteten mündlichen oder schriftlichen Prüfung abgeschlossen. Die Modulendnote wird durch die Note der Prüfung festgelegt. Für die Vergabe der LP gilt die Regelung aus dem Kapitel Prüfungsmodalitäten.	

Useful literature	R. Motwani und P. Raghavan, Randomized Algorithms, Cambridge University Press 1995. M. Mitzenmacher und E. Upfal, Probability and Computing, Cambridge University Press, 1995. N. Alon und J. H. Spencer, The Probabilistic Method, John Wiley and Sons, 2008.

5 Core 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 compulsory optional 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

<table>
<tr><td>Code
MM12</td><td colspan="2">Name
Grundmodul Angewandte Analysis und Modellierung</td></tr>
<tr><td>CP
8 pro Veranstaltung</td><td>Duration
pro Veranstaltung: ein Semester</td><td>Offered
mindestens jährlich</td></tr>
<tr><td>Format
pro Veranstaltung: Vorlesung 4 SWS + Übung 2 SWS</td><td>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</td><td>Availability
Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden.
M.Sc. Mathematik,
M.Sc. Scientific Computing</td></tr>
<tr><td>Language
Deutsch oder Englisch</td><td>Lecturer(s)
wechselnd</td><td>Examination scheme</td></tr>
<tr><td>Learning objectives</td><td colspan="2">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</td></tr>
</table>

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, Beispiele weitere nichtlineare 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
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 im LSF oder auf der Homepage der Vorlesung angegeben

Basic course Numerical Analysis and Optimization

<table>
<tr><td>Code
MM15</td><td colspan="2">Name
Basic course Numerical Analysis and Optimization</td></tr>
<tr><td>CP
8 pro Veranstaltung</td><td>Duration
pro Veranstaltung: ein Semester</td><td>Offered
mindestens jährlich</td></tr>
<tr><td>Format
pro Veranstaltung: Vorlesung 4 SWS + Übung 2 SWS</td><td>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</td><td>Availability
Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden.
M.Sc. Mathematik,
M.Sc. Scientific Computing</td></tr>
<tr><td>Language
Deutsch oder Englisch</td><td>Lecturer(s)
wechselnd</td><td>Examination scheme</td></tr>
<tr><td>Learning objectives</td><td colspan="2">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</td></tr>
</table>

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 im LSF oder auf der Homepage der Vorlesung angegeben

Grundmodul Statistik und Wahrscheinlichkeitsrechnung

Code MM16	**Name** Grundmodul Statistik und Wahrscheinlichkeitsrechnung	
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**
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) Statistik II: Asymptotische Statistik (asymptotische Normalität, Effizienz, Abstandsmaße, Modell-Fehlspezifikation, Tests von Hypothesen); Nichtparametrische Statistik (Nichtparametrische Schätzer, Regularisierung, Konvergenzraten, Kernschätzer, Adaptivität, nichtparametrische Tests); Statistik für komplexe Systeme (z.B. Statistik stochastischer Prozesse, inverse Probleme, hochdimensionale Statistik, Statistik bei Netzwerken)	
Requirements for participation	empfohlen sind Kenntnisse der Analysis und linearen Algebra, Wahrscheinlichkeitstheorie 1 und Statistik 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 im LSF oder auf der Homepage der Vorlesung angegeben

Aufbaumodul Angewandte Analysis und Modellierung

<table>
<tr><td colspan="1">Code
MM22</td><td colspan="2">Name
Aufbaumodul Angewandte Analysis und Modellierung</td></tr>
<tr><td>CP
8 pro Veranstaltung</td><td>Duration
pro Veranstaltung: ein Semester</td><td>Offered
mindestens jährlich</td></tr>
<tr><td>Format
pro Veranstaltung: Vorlesung 4 SWS + Übung 2 SWS</td><td>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</td><td>Availability
Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden.
M.Sc. Mathematik
M.Sc. Scientific Computing</td></tr>
<tr><td>Language
Deutsch oder Englisch</td><td>Lecturer(s)
wechselnd</td><td>Examination scheme</td></tr>
<tr><td>Learning objectives</td><td colspan="2">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.</td></tr>
<tr><td>Learning content</td><td colspan="2">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</td></tr>
<tr><td>Requirements for participation</td><td colspan="2">Grundmodul Angewandte Analysis und Modellierung</td></tr>
</table>

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 im LSF oder auf der Homepage der Vorlesung angegeben

Advanced course numerical analysis and optimization

<table>
<tr><td>Code
MM25</td><td colspan="2">Name
Advanced course numerical analysis and optimization</td></tr>
<tr><td>CP
8 pro Veranstaltung</td><td>Duration
pro Veranstalung: ein Semester</td><td>Offered
mindestens jährlich</td></tr>
<tr><td>Format
pro Veranstaltung: Vorlesung 4 SWS + Übung 2 SWS</td><td>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</td><td>Availability
Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden.
M.Sc. Mathematik
M.Sc. Scientific Computing</td></tr>
<tr><td>Language
Deutsch oder Englisch</td><td>Lecturer(s)
wechselnd</td><td>Examination scheme</td></tr>
<tr><td>Learning objectives</td><td colspan="2">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.</td></tr>
</table>

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 Unterraumkorrekturverfahren, 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 Versuchsplanung, Numerische Lösung mit SQP-Verfahren, effiziente Ableitungsberechnung), Globalisierung der Konvergenz bei Newton-Verfahren für sehr nichtlineare Probleme (Abstiegsstrategien, Natürliche Niveaufunktionen, 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) * Uncertainty Quantification 2: Im Rahmen dieser Veranstaltung werden methodische Ansätze vermittelt, die die Quantifizierung von Unsicherheiten im Zusammenhang mit Differentialgleichungen ermöglichen. Folgende Schwerpunkte werden u.a. behandelt: Karhunen-Loève Expansion, Kollokation bzw. hochdimensionale Quadratur und Interpolation, Dünne Gitter, Stochastische Galerkin Diskretisierung für partielle Differentialgleichungen, Bayessche Formulierung inverser Probleme * Informationsgeometrie und Maschinelles Lernen: - Differentialgeometrie: Mannigfaltigkeiten, Untermannigfaltig- keiten, 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
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 im LSF oder auf der Homepage der Vorlesung angegeben

Aufbaumodul Statistik und Wahrscheinlichkeitsrechnung

Code	**Name**	
MM26	Aufbaumodul Statistik und Wahrscheinlichkeitsrechnung	
CP	**Duration**	**Offered**
8 pro Veranstaltung	pro Veranstaltung: ein Semester	mindestens jährlich
Format	**Workload**	**Availability**
pro Veranstaltung: Vorlesung 4 SWS + Übung 2 SWS	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	Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden. M.Sc. Mathematik M.Sc. Scientific Computing
Language	**Lecturer(s)**	**Examination scheme**
Deutsch oder Englisch	wechselnd	
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: 1. Fortgeschrittene Zeitreihenanalyse 2. Statistik zeitstetiger Prozesse 3. Angewandte Statistik 4. Lokale asymptotische Normalität und Semiparametrik: Asymptotische Entscheidungstheorie für lokal asymptotisch normale Experimente, Differenzierbarkeit im quadratischen Mittel, Kontiguität, Semiparametrik, asymptotische Effizienz in semiparametrischen Modellen 5. Empirische Prozesse: Glivenko-Cantelli Sätze, Vapnik-Cervonenskis Theorie, Konzentrationsungleichungen für empirische Prozesse, Donsker Theoreme, Entropieabschätzungen für Funktionenklassen, Konvergenzraten in der Nichtparametrik 6. Nichtparametrische Minimaxtheorie 7. 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 8. Bayesstatistik 9. 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 im LSF oder auf der Homepage der Vorlesung angegeben

Spezialisierungsmodul Angewandte Analysis und Modellierung

<table>
<tr><td>Code
MM32</td><td colspan="2">Name
Spezialisierungsmodul Angewandte Analysis und Modellierung</td></tr>
<tr><td>CP</td><td>Duration
pro Veranstaltung: ein Semester</td><td>Offered
mindestens jährlich</td></tr>
<tr><td>Format</td><td>Workload</td><td>Availability
Es können mehrere verschiedene Veranstaltungen in diesem Modul absolviert werden.
M.Sc. Mathematik
M.Sc. Scientific Computing</td></tr>
<tr><td>Language
Deutsch oder Englisch</td><td>Lecturer(s)
wechselnd</td><td>Examination scheme</td></tr>
<tr><td>Learning objectives</td><td colspan="2">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.</td></tr>
<tr><td>Learning content</td><td colspan="2">Aktuelle Forschungsthemen aus den Arbeitsgebieten der Dozierenden.</td></tr>
<tr><td>Requirements for participation</td><td colspan="2">empfohlen sind Veranstaltung(en) aus dem Aufbaumodul Analysis und Modellierung</td></tr>
<tr><td>Requirements for the assignment of credits and final grade</td><td colspan="2">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.</td></tr>
<tr><td>Useful literature</td><td colspan="2">wird im LSF oder auf der Homepage der Vorlesung angegeben</td></tr>
</table>

Special topics of numerical analysis and optimization

Code MM35	**Name** Special topics of numerical analysis and optimization	
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**
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 im LSF oder auf der Homepage der Vorlesung angegeben	

Spezialisierungsmodul Statistik und Wahrscheinlichkeitsrechnung

Code MM36	**Name** Spezialisierungsmodul Statistik und Wahrscheinlichkeitsrechnung	
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**
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 im LSF oder auf der Homepage der Vorlesung angegeben	

6 General Modules

This chapter concludes the descriptions of the mandatory modules *Seminar*, *Field of Application* and *Master Thesis* as ewll as the descriptions for the elective modules *Advanced Software Practical* and *Lab internship in field of application.*

During the studies three regular seminars must be completed:
– a seminar in mathematics
– a seminar in computer science (can be replaced by the module *Advanced Software Practical*)
– a master seminar

The master seminar is a regular seminar in mathematics or computer science which must be chosen following the guidelines of your thesis supervisor. Usually the supervisor will ask you to attend one of his regular seminars - or the supervisor can point you to a suitable other seminar. The topic of this seminar should be chosen in preparation of your thesis, but the master seminar is not a seminar about your thesis research.
A possibility to receive the CP in the Application field is the new *Lab internship in field of application.*

Seminar

Code SCS	**Name** Seminar	
CP 6	**Duration** one semester	**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	Recommended prior knowledge 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 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		

Advanced Software Practical

Code SCASP	**Name** Advanced Software Practical	
CP 8	**Duration** one semester	**Offered** every semester
Format 6 SWS Practical	**Workload** 240h; thereof at least 25 h presence 10 h preparation presentation	**Availability** M.Sc. Scientific Computing
Language English or German	**Lecturer(s)** depending on teaching offer	**Examination scheme**
Learning objectives	Students acquire problem solving competence for complex design and implementation tasks and deepen their programming ability.	
Learning content	Carry out a project from concept to execution and presentation. Specific contents depends on the problem to be solved. General approach of a software practical includes - Execution and evaluation of projects and their phase structure - Planing of team work - Presentation of project results	
Requirements for participation		
Requirements for the assignment of credits and final grade	The module is completed with a graded exam. This exam includes the documented software, the project report and the presentation of the results. More details will be announced by the lecturer. The final grade of the module is determined by the grade of the exam.	
Useful literature		

Field of Application

Code SCAP	Name Field of Application	
CP 18	**Duration**	**Offered**
Format	**Workload** 540 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

Code SCLI	**Name** Lab internship in field of application	
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		

Master Thesis

Code SCMa	**Name** Master Thesis	
CP 30	**Duration** one semester	**Offered**
Format	**Workload** 900 h; thereof 810 h working on a thesis topic (research and development tasks) 90 h for preparing and giving a thesis presentation	**Availability** M.Sc. Scientific Computing
Language English or German	**Lecturer(s)** depending on offer	**Examination scheme**
Learning objectives	Supervised research work in a specific research area of Scientific Computing	
Learning content	Research work on a specific topic in Scientific Computing	
Requirements for participation	According to the examination regulations, §14.2, the amount of 45 credits have to be completed and registered before the master thesis can be registered.	
Requirements for the assignment of credits and final grade	Passing the graded master's thesis is required for the award of the CP. Composite grade: 90% thesis (grade given by the two thesis examiners), 10% presentation (grade given by the two examiners)	
Useful literature	Given by supervisor.	

7 Key Competence Program

In this chapter the module descriptions the modules of the Key Competence Program are listed.

This list is still in progress.

The course *C++ Practice* from the MSc Technische Informatik can also be chosen.

Einführung in das Textsatzsystem LaTeX

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

Useful literature	