

Module Handbook for the Master's Degree in Computer Science

Institute of Computer Science
of the Faculty of Mathematics and Natural Sciences
at the Heinrich Heine University

Published by the
Committee for the Master Examination
in Computer Science

Updated on 16.02.2024

Foreword

The module handbook is intended to provide orientation for the basic Bachelor's degree in Computer Science and the consecutive Master's degree in Computer Science. In particular, it should make it easier to choose courses and support the organization of the course.

The module handbook lists the usual courses. However, it is not a complete, exhaustive or definitive listing. Rather, the module handbook is continuously updated and thus reflects the development in research and teaching at the Institute of Computer Science.

In the section on participation requirements, we distinguish between formal and content-related requirements. If you do not meet the formal requirements, you may not register for this module. Modules with content requirements require certain knowledge. If you do not have this knowledge, you will have to acquire it yourself. In general, we recommend completing the relevant modules.

Please note, however, that the respective bachelor's or master's examination regulations for the subject of computer science are decisive for all questions relating to studies and examinations.

Düsseldorf, 16.02.2024

The Committee for the Master Examination in Computer Science

Inhaltsverzeichnis

Foreword	2
Qualification goals of the M.Sc. in Computer Science	5
Scientific qualifications and qualifications for employment	5
Personal development	5
Course Plan	6
Courses for elective areas	7
Advanced topics in Bayesian Data Science	8
Algorithms for Ad-hoc and Sensornetworks	10
Algorithmics for Hard Problems	12
Algorithms and Data Structures 2	14
Algorithmic Game Theory	16
Algorithms for Sequence Analysis	18
Approximation Algorithms	20
Approximation Algorithms for Clustering Problems	22
Operating System Development	24
Create Your Tech Startup	26
Discrete Optimization: Theory and Applications	28
Dynamic Programming Languages	30
Introduction to Statistical Analysis through Computer Simulations	32
Fair Division	34
Algorithms for Graphs 2	36
Growth Mechanics	38
Information Theory	40
Isolation and Protection in Operating Systems	42
Knowledge Discovery in Databases - Selected Topics	44
Knowledge Discovery in Databases	46
Cryptocomplexity 2	48
Master's Seminar: Current Topics in Theoretical and Computational Biology	50
Master's Seminar: Algorithms for Graph Theoretical Concepts in Computer Science	51
Master's Seminar: Algorithms for Perfect Graphs	52
Master's Seminar: Operating System Development with Rust	54
Master's Seminar: Combinatorial Optimization in Bioinformatics	56
Master's Seminar: Computational Multiomics	58
Master's Seminar: Databases and Information Systems	60
{Master's Seminar: Deep Learning	61
Master's Seminar: Digital Innovation and Entrepreneurship	63
Master's-Seminar: Collective Decision-Making	65
Master's-Seminar: Complexity and Cryptology	67
Master's Seminar on NP-hard problems	69
Master's Seminar: Property Testing	71
Master's Seminar: Reproducibility in Bioinformatics Research	73
Master's Seminar: User Experience (UX) Design	75
Master's Seminar: Scientific Methods	77
Methods for Population Genetics	79

Model Checking	81
Multimedia Database Systems	82
Natural Language Processing	84
Practical: Implementing Transformer Models	86
Representations in Natural Language Processing	88
Safety-Critical Systems	90
Stochastic Models of Biological Systems	92
System Software for Big Data Computing	94
Transaction Management	95
User Experience (UX) Design and Management	97
Advanced Compiler Construction	99
Advanced Functional Programming: Clojure	101
Advanced Topics in Logic Programming	103
Courses from other programs or institutes	105
Master AI and Data Science	105
Master Physik	106
Master Betriebswirtschaftslehre	107
Individual Research Project	108
Module Description	109
Master's Thesis	110
Module Description	110
Modules no longer offered	111
Big-Data Systems and Applications	112
Introduction to Linear Optimization	114
Judgment Aggregation	116
Master's Seminar: System Software for Big Data Computing	118

Qualification goals of the M.Sc. in Computer Science

Upon completion of their studies, students will have achieved the following qualification goals:

Scientific qualifications and qualifications for employment

Graduates

- are able to independently compare and apply complex methods and procedures of computer science
- can independently analyse, evaluate and apply scientific methods
- have dealt with currently research topics in theory and/or practice and are able to solve problems and evaluate their results
- have shown that they can independently perform research in which they have applied learned knowledge using research methods to a derive research question, taking into account generally accepted principles of good scientific practice, and have assessed its usefulness.

Personal development

Graduates

- have acquired deeper knowledge in their selected focus modules
- are able to communicate and discuss complex computer science problems and appropriate solutions taking into account others' points of views.
- can assess their own skills and have concrete ideas for their further professional development. They can also independently acquire extensive specialised knowledge.
- are aware of ethical questions and challenges from the perspective of computer science as well as the social, cultural and political significance of their discipline.

Course Plan

You can find the course plan on the [German website of the Institute of Computer Science](#). It is called **Studienverlaufspläne** and serves as orientation and help you to plan your studies.

Courses for elective areas

In the Computer Science master's program, teaching units (modules) of different sizes are offered for the compulsory elective areas *Practical or Technical Computer Science*, *Theoretical Computer Science*, *Individual Specialization* and *Focus*. The number of credit points of the modules was adjusted for the winter semester 2015/2016. This conversion took place as part of the adjustment of the study regulations for the master's degree in computer science to the framework study regulations for master's degree programs in the Faculty of Mathematics and Natural Sciences.

When registering for the exam, the student specifies which elective area he would like to assign the module to. The compulsory elective areas of a module that can be selected are listed under "Usability of the module".

The selection of the modules for the compulsory elective area "focus" must be discussed with the mentor and supervisor of the master's thesis (see form "choice of focus").

The *Bioinformatics* focus area is open to both students who already have previous knowledge of bioinformatics and those who have studied computer science in their bachelor's degree with a focus other than bioinformatics. Therefore, the course *Algorithms in bioinformatics* from the bachelor's degree is also offered to master's students without previous knowledge of bioinformatics. A differentiation between master and bachelor students is made via a separate seminar task.

The language of instruction is either German or English. If no information is given on the language of instruction in the module descriptions, the language of instruction is German. However, knowledge of English is essential for successful studies. Even if events are held in German, the written materials used are often in English.

Oral examinations are conducted in German or English. However, English can only be chosen as the examination language if everyone involved in the examination agrees.

The written theses are written in German or English.

The formal requirement for participation in courses on modules that are only offered for the Master's degree is the successful completion of the modules *Programming*, *Computer Architecture*, *Algorithms and Data Structures* as well as *Theoretical Computer Science*. Summarized below as *Prerequisites for anticipating Master's modules*.

Advanced topics in Bayesian Data Science

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	every winter semester	Master Computer Science	English

Content

- Introduction to Bayesian hierarchical models, Bayesian inference and empirical Bayesian inference, Bayesian Monte Carlo techniques and MCMC calculations
- Introduction to Bayesian meta-analysis, Bayesian meta-analysis for diagnostic test data, Bayesian evidence synthesis for deterministic models
- Exploratory data analysis of multilevel and longitudinal data, Bayesian hierarchical linear models, Bayesian hierarchical generalized linear models
- Bayesian models for statistical regression problems, Bayesian linear regression with a large number of covariates, hierarchical pre-distributions and variable selection
- Bayesian models for exploratory multivariate analysis, a priori distributions for variance-covariance matrices, multivariate comparisons of groups and other multivariate methods
- Bayesian methods to deal with missing data, Bayesian approaches to missing random data, Bayesian approaches to missing non-random data
- Bayesian methods of censoring and truncation, modeling survival and time to event data, handling truncated data
- Bayesian non-parametric models, introduction to mixture models, statistical modeling with Dirichlet process mixtures

Learning Outcomes

After completing the course, students are able to

- describe the difference between Empirical Bayes (EB) and full Bayesian analysis in hierarchical models,
- compare EB and full Bayesian approaches in multi-level regression, logistic regression and multivariate analysis,
- perform data analysis for aggregated data using Bayesian hierarchical models,
- develop Bayesian hierarchical models for multi-level and longitudinal data,
- develop scripts in R, OpenBUGS and JAGS software for hierarchical Bayesian statistical modelling,
- describe different Bayesian hierarchical models for different types of outcomes (e.g. continuous, categorical, ordinal categorical, time-to-event),
- perform Bayesian non-parametric data analysis for different types of data outcomes,
- characterize different types of missing data (MCAR, MAR, MNAR) and different types of modeling missing data (e.g. not-ignorability missing mechanisms), and
- implement and develop Bayesian hierarchical models in new situations of data analysis.

Bibliography

- Andrew Gelman, John B. Carlin, Hal S. Stern: Bayesian Data Analysis. Chapman & Hall. 2013.
- David Lunn, Chris Jackson, Nicky Best, Andrew Thomas, David Spiegelhalter. The BUGS Book (Chapman & Hall/CRC Texts in Statistical Science). 2012

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics
- Elective Area Master study programme Artificial Intelligence and Data Science

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Contents of module *Introduction to Statistical Analysis through Computer Simulations* or a very good background knowledge in the fields of probability theory, statistics and the use of the statistical software R and OpenBUGS or JAGS.

Conditions for awarding credit points

- active participation in the exercises
- successful completion of the exercises
- passing the final exam (usually written)

Responsible persons

Prof. Dr. Martin Lercher, Dr. Pablo Verde

Algorithms for Ad-hoc and Sensornetworks

ECTS credits	Total hours	Contact hours	Self-study hours
10 CP	300 hours	90 hours	210 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (4 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Remarks

- 15 CP for students studying according to PO 2005.

Content

This module imparts basic knowledge from the following areas:

- distributed algorithms
- leader election
- geographic routing
- topology control
- location based services
- location determinations
- greedy embeds
- beacon routing
- interval routing
- hop networks
- network coding
- distributed computation for dominant sets
- distributed computation for maximum independent sets

Learning Outcomes

After completing the course, students are able to

- formulate distributed calculations in ad hoc networks
- perform various routing techniques in ad hoc networks
- apply topology control to ad hoc networks
- determine locations and virtual addresses in ad hoc networks

Bibliography

- Dorothea Wagner und Roger Wattenhofer: Algorithms for Sensor and Ad Hoc Networks. Springer, LNCS 4621. Berlin, 2007.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- active participation in the exercises,
- submission of selected homework,
- written exam (usually 90 minutes) or oral exam at the end of the semester

Responsible persons

Prof Dr. Egon Wanke

Algorithmics for Hard Problems

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Remarks

- 7.5 CP for students studying according to PO 2005.

Content

This module deals with hard problems for which there are probably no efficient algorithms and presents, among others, the following approaches to solve such problems.

- pseudopolynomial algorithms
- algorithms on special graphs (trees and co-graphs)
- parameterized algorithms (standard parameters, parameterization by tree-width, parameterization by clique-width)
- exact exponential time algorithms

Learning Outcomes

After completing the course, students are able to

- explain and formally define the hard problems discussed,
- explain the algorithms discussed for hard problems and apply them to concrete inputs,
- perform a parameterized runtime analysis of algorithms and
- construct the considered tree structures for given instances.

Bibliography

- F. Gurski, I. Rothe, J. Rothe, E. Wanke: Exakte Algorithmen für schwere Graphenprobleme. Springer Verlag. 2010.
- F.V. Fomin, D. Kratsch: Exact Exponential Algorithms. Springer Verlag. 2010.
- J. Hromkovic: Algorithmics for Hard Problems. Springer Verlag. 2003.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formally: Bachelor's students must meet the requirements for the anticipation of Master's modules.

Conditions for awarding credit points

- active participation in the exercises
- handing in the exercises
- passing the written exam at the end of the semester

Responsible persons

Priv.-Doz. Dr. Frank Gurski

Algorithms and Data Structures 2

ECTS credits	Total hours	Contact hours	Self-study hours
10 CP	300 hours	90 hours	210 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (4 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Content

This module imparts basic knowledge from the following areas:

- concatenate and split of balanced search trees
- median of medians
- priority queues (left trees, binomial queues, Fibonacci heaps, amortized runtime analysis)
- algorithmic design techniques (greedy, divide and conquer, dynamic programming)
- low memory algorithms (in place models, read only models, streaming models, min count sketches)
- preprocessing (minimum range requests, cartesian trees, Fischer Heun scheme, 4 russians method)
- algorithmic geometry (maximum area triangles in polygons, minimum enclosing circles)
- dynamic algorithms (connectivity of graphs, convex hull)
- parallel Algorithms (PRAM model, network model, parallel complexity classes, parallel basic algorithms)
- parameterized Algorithms (fixed parameter tractability, W hierarchy)

Learning Outcomes

After completing the course, students are able to

- use advanced theoretical and practical concepts of complex algorithms and data structures,
- design and analyze efficient algorithms and data structures for combinatorial problems from different areas,
- apply a wide range of advanced techniques and strategies to solve algorithmic problems, and
- classify and evaluate current research topics in the field of algorithms.

Bibliography

- Selected publications regarding the topic of the course.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- active participation in the exercises,
- submission of selected homework,

- written exam (usually 90 minutes) or oral exam at the end of the semester

Responsible persons

Prof Dr. Egon Wanke

Algorithmic Game Theory

ECTS credits	Total hours	Contact hours	Self-study hours
10 CP	300 hours	90 hours	210 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (4 HPW) Tutorial (2 HPW)	irregular	Bachelor Computer Science	German

Remarks

- 15 CP for students studying according to PO 2005.

Content

This module provides a basic understanding of the foundations, models, methods, and results from the area of algorithmic game theory. One distinguishes between noncooperative and cooperative games and studies, in particular, equilibria and stability concepts to predict the outcome of a game. The related problems are investigated in terms of their algorithmic and complexity-theoretic properties.

Noncooperative Games: Playing Against Each Other

- Foundations (Normal Form, Dominant Strategies and Equilibria, Two-Person Games)
- Nash Equilibria in Mixed Strategies (Definition and Properties; Existence)
- Checkmate: Game Trees in Games with Perfect Information (Sequential Two-Person Games; Equilibria in Game Trees)
- Full House: Games with Incomplete Information (the Goat Problem; Analysis of a Simple Poker Variant)
- How Hard Is It to Find a Nash Equilibrium? (Nash Equilibria in Zero-Sum Games and in General Normal Form Games)

Cooperative Games: Playing with Each Other

- Foundations (Cooperative Games with Transferable Utility; Superadditive Games; Stability Concepts for Cooperative Games)
- Simple Games (the Core of Simple Games; Representations of Simple Games; Weighted Voting Games; Dimensionality; Power Indices; the Shapley-Shubik Index and the Shapley Value; the Banzhaf Indices)
- Complexity of Problems for Compactly Representable Games (Games on Graphs; Weighted Voting Games; Hedonic Games)

Learning Outcomes

After completing the course, students are able to

- describe and explain the considered two-person games in normal form and their properties (dominant strategies, Nash equilibria, Pareto optimality, etc.),
- calculate Nash equilibria in mixed strategies,
- describe the most important steps in Nash's Theorem (awarded with a Nobel Prize) about the existence of equilibria in mixed strategies,
- determine equilibria in game trees,
- analyze the considered simple poker variant,
- describe the complexity of problems related to Nash equilibria,
- describe and explain the considered cooperative games and their properties,
- describe the representation and properties (in particular, stability concepts) of simple games (e.g., whether they have an empty core),
- describe and calculate the power indices in weighted voting games, and

- describe the complexity of the considered problems for weighted voting games, games on graphs, and hedonic games.

Bibliography

- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division. Springer Texts in Business and Economics, Springer-Verlag. Heidelberg, 2015. 1. Auflage.
- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen. Spektrum Akademischer Verlag (Springer). Heidelberg, 2011. 1. Auflage.
- Further literature is provided during the course.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- active and successful participation in the exercises
- written exam (exam, usually 90 minutes)

Responsible persons

Prof. Dr. Jörg Rothe

Algorithms for Sequence Analysis

ECTS credits	Total hours	Contact hours	Self-study hours
10 CP	300 hours	90 hours	210 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (4 HPW) Tutorial (2 HPW)	every summer semester	Master Computer Science	English

Content

This lecture addresses classic as well as recent advanced algorithms for the analysis of sequences, with an emphasis on algorithms that are fast in practice. DNA sequencing data are one example that motivates this lecture, but the focus of this course is on algorithms and concepts that are not specific to bioinformatics.

- Full text search without index
- Approximate pattern matching
- Suffix trees, enhanced suffix arrays, and linear-time construction algorithms
- Burrows-Wheeler Transform (BWT) and the FM index
- De Bruijn graphs and overlap graphs
- Data compression
- Advanced pairwise alignment algorithms
- Multiple sequence alignment
- Positional Burrows-Wheeler Transform (PBWT)
- Locality sensitive hashing

Learning Outcomes

After completing the course, students are able to

- state concepts for different data structures to index strings
- explain algorithms to build these data structures
- explain different algorithms for text search and sequence alignment
- design and implement advanced algorithms for sequence analysis
- analyze and discuss the merits of different algorithms both in terms of asymptotic runtime and practical performance

Bibliography

- Ohlebusch: Bioinformatics Algorithms. Oldenbusch Verlag. 2013.
- Navarro and Raffinot: Flexible Pattern Matching in Strings. Cambridge University Press. 2002.
- Mäkinen, Belazzougui, Cunial, Tomescu: Genome-Scale Algorithm Design. Cambridge University Press. 2015.
- Gusfield: Algorithms on Strings, Trees, and Sequences. Cambridge University Press. 1997.

Module compatibility

- Elective Area Theoretical Computer Science
- Elective Area Practical or Technical Computer Science
- Elective Area Master study programme Artificial Intelligence and Data Science
- Application subject for Minor area Master study programme Mathematics
- Individual Supplement

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Contents of modules *Programming, Algorithms and Data Structures* und *Theoretical Computer Science*

Conditions for awarding credit points

- The weekly exercise sheets come with programming tasks and theory tasks. To be eligible to take the exam, students have to earn 50% of programming tasks and 50% of theory tasks.
- oral exam (usually 20 minutes)

Responsible persons

Prof. Dr. Tobias Marschall

Approximation Algorithms

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Content

This course is an advanced algorithms lecture which introduces basic concepts from the area of approximation algorithms. Approximation algorithms are algorithms with a polynomial time bound for difficult optimization problems which satisfy a provable guarantee on the quality of the solution. The lecture covers techniques for the design and analysis of approximation algorithms, in particular:

- greedy algorithms
- approximation algorithms for graph problems
- problems with Euclidean vs metric vs non-metric inputs
- (I)LP-based algorithms
- polynomial approximation schemes

Learning Outcomes

After completing the course, students are able to

- explain the concept "approximation ratio",
- formally define the discussed optimization problems,
- suitably summarize and explain complex proofs,
- apply basic techniques for the analysis of approximation ratios and
- apply basic design techniques for approximation algorithms.

Bibliography

- Vijay Vazirani: Approximation Algorithms. Springer-Verlag, Berlin, 2003. 1. Ausgabe
- David P. Williamson and David B. Shmoys: The Design of Approximation Algorithms, Cambridge University Press, Cambridge, 2011, 1. Ausgabe.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formally: Bachelor's students must meet the requirements for the anticipation of Master's modules.

Conditions for awarding credit points

- Active participation in the exercises
- handing in the exercises

- passing the written exam at the end of the semester

Responsible persons

Prof. Dr. Melanie Schmidt, Dr. Daniel Schmidt

Approximation Algorithms for Clustering Problems

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	English German

Content

This course is a lecture on advanced algorithms and deals with clustering algorithms. The focus lies on approximation algorithms. In particular, we discuss the following topics:

- LP-based techniques
- local search
- approximation algorithms for the k-median problem and for the k-means problem
- clustering in the Euclidean space
- dimensionality reduction and data stream algorithms for clustering problems

Learning Outcomes

After completing the course, students are able to

- model discussed and new clustering problems as linear programs,
- describe advanced analysis techniques for discussed clustering problems,
- explain discussed algorithmic ideas for clustering problems, and
- analyze the approximation guarantee of discussed and slightly varified clustering procedures.

The language of instruction for the respective semester is announced in HIS/LSF.

Bibliography

- Selected publications regarding the topic of the course.
- Lecture Notes

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: It is helpful to know the contents of the modules *Approximation Algorithms* and *Combinatorial Algorithms for Clustering Problems*

Conditions for awarding credit points

- actively participate in tutorials
- final exam (written or oral exam)

Responsible persons

Prof. Dr. Melanie Schmidt

Operating System Development

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (1 HPW) Tutorial (3 HPW)	every summer semester	Master Computer Science	German

Remarks

- 7.5 CP for students studying according to PO 2005.

Content

The lecture covers basic concepts relevant for the 64 bit operating system development in the exercises. This allows participation of students who have not attended the Bachelor module "Operating Systems and System Programming".

Lecture content: - Boot process - x86_64 programming model - Interrupts (PIC and APIC) - Coroutines and threads - Scheduling - Synchronisation - Driver architectures

The main focus of this module is on the exercises where each student develops from scratch its own x86-based 64 bit operating system. The development is done using either C++ or Rust.

Learning Outcomes

After completing the course, students are able to

- to use either C++ or Rust for hardware programming,
- develop basic operating system functions and
- program and synchronize concurrent tasks (interrupts and threads) in operating systems

Bibliography

- Intel Corporation, Intel Architecture Software Developer's Manual.
- Further literature is provided during the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formally: Bachelor students must meet the requirements for the anticipation of Master modules.

Conditions for awarding credit points

- Exam about the own operating system which must implement the functions of all exercise sheets.

Responsible persons

Prof. Dr. Michael Schöttner

Create Your Tech Startup

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	every summer semester	Master Computer Science	English

Content

Students explore the entrepreneurial process using a learning by doing methodology.

- The lecture and case study sessions provide and discuss tools and methods of creating, visualizing, and analyzing digital business models (e.g., business model canvas, lean startup, design thinking).
- In a group project, students transfer and apply those tools and methods to create, evaluate, plan, and pitch their own tech startup ideas.

Learning Outcomes

After completing the course, students are able to

- model, analyze, and discuss digital business models and its components,
- assess the specific opportunities for and challenges of technology-based businesses,
- create, plan, and implement novel tech startups,
- pitch their startup idea in front of peers and experts,
- present, assess, and give feedback to novel tech business models,
- assess their entrepreneurial skills and
- collaborate with interdisciplinary peers comprising various competences.

Bibliography

- Blank, S.; Dorf, B.: The Startup Owner's Manual: The Step-By-Step Guide for Building a Great Company. K & S Ranch, 2012. 1st edition
- Osterwalder, A; Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons, 2010. 1st edition
- Osterwalder, A; Pigneur, Y.; Bernarda, G; Smith, A.: Value Proposition Design: How to Create Products and Services Customers Want. John Wiley & Sons, 2014. 1st edition
- Ries, E.: The Lean Startup: How Constant Innovation Creates Radically Successful Businesses, Portfolio Penguin, London. 2011. 1st edition
- Further literature is provided during the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics
- Elective Area Master study programme Artificial Intelligence and Data Science

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- Presentations in groups
- Written documentation of a business plan
- Active class participation in discussions

Responsible persons

Prof. Dr. Steffi Haag

Discrete Optimization: Theory and Applications

ECTS credits	Total hours	Contact hours	Self-study hours
10 CP	300 hours	120 hours	180 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (4 HPW) Tutorial (2 HPW)	every winter semester	Master Computer Science	English

Content

The lecture covers the basic theory and applications of (integer) linear and discrete optimization. In the exercises the content of the lecture is applied and deepened. For that the exercises contain theoretical as well as practical elements. The students use linear, integer linear programming and other discrete optimization modeling software and solvers to solve applied programming exercises.

- Foundations of Linear Programming
- Linear Programs and their geometric interpretation
- Duality
- The Simplex method
- Integer Linear Programming
- Linear programming-based Branch-and-Bound
- Cutting planes and Branch-and-Cut
- Network flows
- Satisfiability: Modeling and SAT solvers
- Constraint Programming
- Applications: Selected applications of discrete optimization techniques
- Project work on selected discrete optimization problem

Learning Outcomes

After completing the course, students are able to

- state the basic principles of linear and integer linear programming (LP and ILP) as well as SAT and constraint programming
- build practical LP and discrete optimization models
- design and implement algorithms that solve these models

Bibliography

- Vašek Chvátal: Linear Programming. W.H. Freeman, 1983
- Dimitris Bertsimas, John N. Tsitsiklis: Introduction to Linear Optimization. Athena Scientific. Belmont, MA, USA, 1997.
- Laurence Wolsey: Integer Programming. Wiley. New York, NY, USA, 1998.
- Dan Gusfield: Integer Linear Programming in Computational and Systems Biology. Cambridge University Press, 2019

Module compatibility

- Elective Area Theoretical Computer Science
- Elective Area Practical or Technical Computer Science
- Elective Area Master study programme Artificial Intelligence and Data Science
- Major Subject

- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formally: Bachelor's students must meet the requirements for the anticipation of Master's modules.
- No credit points are possible if the precursor module "Introduction to Linear Programming" has been successfully completed.

Conditions for awarding credit points

- active participation
- successful completion of exercise sheets and project work
- passing written exam

Responsible persons

Prof. Dr. Gunnar W. Klau

Dynamic Programming Languages

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	every summer semester	Master Computer Science	German

Remarks

- 7.5 CP for students studying according to PO 2005.

Content

This lecture conveys typical concepts and characteristics dynamic programming languages, e.g. typing and duck-typing, Metaprogramming, etc. In addition, we discuss how a interpreter for a dynamic programming language is implemented, to support the concepts discussed earlier.

In the practical exercises (self study), students deal with various problems using the properties of a dynamic programming language to gain a deeper understanding of these concepts. In the second half of the semester, the task in the exercise is to independently write an interpreter for a predefined dynamic programming language.

Learning Outcomes

After completing the course, students are able to

- describe the principles of dynamic programming and compare them with classical, imperative programming languages,
- evaluate for which areas of application dynamic programming is advantageous,
- to independently create and test functional programs, and
- independently write an interpreter for a given dynamic programming language.

Bibliography

- Robert Nystrom: *Crafting Interpreters*. Genever Benning. 2021. 1st Edition

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Contents of module *Compiler Construction*

Conditions for awarding credit points

- Successful processing of compulsory exercises

- Successful development of your own interpreter
- Passing the exam

Responsible persons

Dr. John Witulski

Introduction to Statistical Analysis through Computer Simulations

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	every winter semester	Master Computer Science	English

Remarks

- 7.5 CP for students studying according to PO 2005.

Content

- Introduction into Statistical Modeling and Bayesian thinking (Probability and introduction to computer simulations, Introduction to Bootstrapping, Introduction to Bayesian Modeling, Introduction to multivariate probability distributions and multiple parameter models)
- Monte Carlo simulation methods (Monte Carlo method for calculating integrals, Rejection sampling, Importance sampling, Sampling importance re-sampling)
- Markov chain Monte Carlo methods (Introduction to Markov chains, Metropolis-Hastings algorithm, Directed acyclic graphs, Gibbs-sampling, MCMC output analysis)
- Statistic modeling (Regression modeling, Analysis of multiple contingency tables, Introduction to hierarchical models)

Learning Outcomes

After completing the course, students are able to

- represent the role of Bayesian Analysis in a broad spectrum of data analysis,
- describe and use different prior distributions in Bayesian Analysis,
- delineate simple and advanced Monte Carlo simulation methods. In particular the application of Bootstrap methods in classical data analysis and Markov Chain Monte Carlo (MCMC) in Bayesian analysis,
- compare classical and Bayesian approaches in regression, logistic regression and multivariate analysis,
- apply computer simulations in classical and Bayesian data analysis,
- use R, OpenBUGS and JAGS software for Bayesian statistical modeling,
- apply simple Bayesian hierarchical models in data analysis, and
- implement and develop Bayesian models in new situations of data analysis.

Bibliography

- Andrew Gelman, John B. Carlin, Hal S. Stern: Bayesian Data Analysis. Chapman & Hall. 2013.
- David Lunn, Chris Jackson, Nicky Best, Andrew Thomas, David Spiegelhalter. The BUGS Book (Chapman & Hall/CRC Texts in Statistical Science). 2012

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement

- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- active participation in the exercises
- successful completion of the exercises
- passing the final exam (usually a project presentation)

Responsible persons

Dr. Pablo Verde, Prof. Dr. Martin Lercher

Fair Division

ECTS credits	Total hours	Contact hours	Self-study hours
10 CP	300 hours	90 hours	210 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (4 HPW) Tutorial (2 HPW)	irregular	Bachelor Computer Science	German

Content

This module provides a basic understanding of the foundations, models, methods, and results from the area of fair division of goods or resources. One distinguishes between the division of divisible and indivisible goods/resources and studies, in particular, methods that guarantee certain concepts of fairness (such as proportionality or envy-freeness) for each division, no matter which individual valuations the agents have for which parts of the goods/resources to be divided. The division of a divisible resource is also known as cake-cutting. The related problems are also investigated in terms of their algorithmic and complexity-theoretic properties.

Cake-Cutting: Fair Division of Divisible Goods

- Foundations
- Valuation Criteria (Fairness; Efficiency; Manipulability; Runtime)
- Two Envy-Free Cake-cutting Protocols for Two Players
- Proportional and Superproportional Cake-cutting Protocols for n Players
- Unequal Shares
- Envy-Free Protocols for Three and Four Players
- Dirty-Work Protocols
- Minimizing the Number of Cuts
- Degree of Guaranteed Envy-Freeness

Fair Division of Indivisible Goods

- Definition and Classification of Allocation Problems
- Preference Elicitation and Compact Representation
- Pareto Efficiency and Envy-Freeness
- Relaxations of Envy-Freeness
- Maximizing Social Welfare: The Santa Claus Problem
- Centralized Fair Division with Ordinal Preferences
- Fair Division with Money, and Related Issues
- Decentralized Allocation Protocols

Learning Outcomes

After completing the course, students are able to

- describe the considered valuation criteria for the studied protocols and explain them for examples,
- unassistedly apply the studied protocols and determine the resulting divisions as well as their properties,
- describe the relations among the presented properties (e.g., Pareto efficiency versus envy-freeness),
- describe the most central results of this course and explain the argumentation in the proofs for examples,
- describe the presented methods of preference elicitation and compact representation for instances of the division of indivisible goods,
- explain relaxations of envy-freeness for examples and
- describe the complexity of problems of maximizing social welfare (in particular, the Santa Claus

Problem).

Bibliography

- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division. Springer Texts in Business and Economics, Springer-Verlag. Heidelberg, 2015. 1. Auflage.
- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen. Spektrum Akademischer Verlag (Springer). Heidelberg, 2011. 1. Auflage.
- Further literature is provided during the course.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- active and successful participation in the exercises
- written exam (exam, usually 90 minutes)

Responsible persons

Prof. Dr. Jörg Rothe

Algorithms for Graphs 2

ECTS credits	Total hours	Contact hours	Self-study hours
10 CP	300 hours	90 hours	210 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (4 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Remarks

- 15 CP for students studying according to PO 2005.

Content

This module imparts basic knowledge from the following areas:

- planar graphs (detection, duality, coloring problems, ...)
- chordal graphs (lexicographic breadth-first search, perfect elimination order, ...)
- tree-width bounded graphs
- clique width, rank width, NLC width
- monadic second-order logic for relational graph structures
- dynamic programming on tree-structured graphs
- minor theorem
- extremal graph theory
- distance hereditary graphs / cographs

Learning Outcomes

After completing the course, students are able to

- classify different graph classes among each other and determine their recognition complexity
- compare width parameters of graph classes
- express graph problems with logical formulas for relational graph structures
- define graph classes via graph operations

Bibliography

- Krishnaiyan „KT” Thulasiraman, Subramanian Arumugam , Andreas Brandstädt , Takao Nishizeki: Handbook of Graph Theory, Combinatorial Optimization, and Algorithms. Chapman & Hall/CRC, Computer & Information Science Series. 2012.
- Selected publications regarding the topic of the course.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- active participation in the exercises
- submission of selected homework
- written exam (usually 90 minutes) or oral exam at the end of the semester

Responsible persons

Prof Dr. Egon Wanke

Growth Mechanics

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	every semester	Master Computer Science	English

Content

With the help of selected examples, the course describes the modeling and analysis of cellular reaction networks, combining multidisciplinary methods from Mathematics, Biology, Physics, Chemistry, and Computer Science.

- Introduction to the mathematical modeling and analysis of cellular reaction networks as an optimal resource allocation problem
- The main physical constraints on cellular growth and replication
- Growth modeling: self-replicator models and nonlinear optimization
- Growth Economy and Control Analysis: the marginal value of reactions and specific costs and benefits
- Growth Balance Analysis: the analytical conditions for optimal cellular growth
- Principles of nonlinear numerical optimization

Learning Outcomes

After completing the course, students are able to

- describe the principles of mathematic modeling of cellular reaction networks,
- can give various examples on how physical constraints on cellular growth and replication can be modeled,
- build and analyze self-replicator cellular models using the GNU R or Python programming languages,
- describe basic concepts behind linear and nonlinear numerical optimization and
- highlight the specific differences and computational challenges of these concepts.

Bibliography

- Relevant literature is provided during the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics
- Elective Area Master study programme Artificial Intelligence and Data Science

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Students should be comfortable with the fundamentals of multivariate Calculus and Linear Algebra, and should be proficient with a programming language.

Conditions for awarding credit points

- at least 50% of the points from the exercises
- passing final exam (written exam, usually 90 minutes)

Responsible persons

Dr. Hugo Dourado, Prof. Dr. Martin Lercher

Information Theory

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	English

Content

In this module students obtain a a better understanding of the concepts of information and randomness, as captured by the concepts of entropy, information loss and Kolmogorov complexity. The relation to coding, data compression and signal transmission is worked out. They are led to a principled view on general assumptions and techniques in machine learning and statistics, like the information bottleneck or the maximum likelihood method.

- What is information?
- Entropy, conditional, joint and relative entropy, mutual information and their interrelations,
- relation to coding, Source Coding Theorem
- characterizations of entropy and relative entropy
- Kolmogorov complexity and connection to entropy, applications of Kolmogorov complexity
- entropy of stochastic processes, channel capacity and Channel Coding Theorem
- differential entropy, Gaussian channel
- maximum entropy distributions
- information theory and statistics: maximum likelihood and exponential families
- applications

Learning Outcomes

After completing the course, students are able to

- define and relate the basic notions of information theory,
- judge the plausibility of previously unseen relations between these notions and if true prove them,
- carry out coding algorithms and explain what they achieve and
- use information theoretic principles to analyze and justify methods of machine learning and statistics

Bibliography

- Cover, Thomas: Elements of Information Theory. Wiley-Interscience. 2012. 2nd ed.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- successful and active participation in the exercise classes
- passing the exam (written or oral exam)

Responsible persons

Dr. Peter Arndt

Isolation and Protection in Operating Systems

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (1 HPW) Tutorial (3 HPW)	every winter semester	Master Computer Science	German

Content

In the lecture, the focus is on isolation and protection through process address spaces. Techniques for separating logical address spaces (page-based and segment-based) as well as accessing beyond address space boundaries are discussed. In this context, various operating system architectures are compared, and common address space models are discussed. Additional topics include interprocess communication through messaging and shared memory.

Selected lecture contents are practically applied in the exercises using Rust (or C++). Starting from a single-core x86-based 64 bit operating system, privilege isolation, spatial isolation, and advanced address space concepts are implemented. The initial operating system is provided or alternatively, the own operating system from the *Operating System Development* module can be used.

Learning Outcomes

After completing the course, students are able to

- use the programming language Rust (or C++) for system-level programming,
- explain and implement design principles for system calls,
- implement basic mechanisms for virtual and physical memory management,
- explain, compare, and develop protection mechanisms in an operating system,
- compare and program mechanisms and abstractions for interprocess communication, and
- describe and compare operating system architectures.

Bibliography

- Intel Corporation, Intel Architecture Software Developer's Manual.
- Further literature is provided during the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- Exam about the own operating system which must implement the functions of all exercise sheets.

Responsible persons

Prof. Dr. Michael Schöttner

Knowledge Discovery in Databases - Selected Topics

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Remarks

- 7.5 CP for students studying according to PO 2005.

Content

Based on the knowledge from the module “Knowledge Discovery in Databases”, further methods are introduced here. The idea of this module is to focus on selected topics in order to reflect current developments in research. Possible topics are for instance:

- temporal association rule mining
- time series data: outlier detection, clustering, and finding frequently recurrent patterns
- further clustering methods (e.g. subspace clustering)
- cluster validity indexes
- clustering with missing values

Learning Outcomes

After completing the course, students are able to

- assess concrete algorithms within the considered topics and to carefully compare such algorithms, and
- select appropriate methods based on the characteristics of the problem to be solved and, if required, combine methods.

Bibliography

- None

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Contents of module *Knowledge Discovery in Databases*

Conditions for awarding credit points

- active participation in the exercises
- homeworks

- final examination (usually as oral examination)

Responsible persons

Prof. Dr. Stefan Conrad

Knowledge Discovery in Databases

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Remarks

- 7.5 CP for students studying according to PO 2005.

Content

This module considers different kinds of knowledge discovery (resp. data mining or machine learning) as well as basic algorithms and discusses the properties of these algorithms. Main focus areas are clustering methods, classification algorithms and methods for association rule mining. In addition, we deal with techniques for efficiently applying KDD methods to large amounts of data. The content in detail is:

- foundations of statistics
- clustering (partitioning, density-based and hierarchical methods)
- classification (evaluating classification algorithms; Bayes, kNN, decision tree, support vector machine)
- association rules (frequent itemsets, Apriori-algorithm, efficient storage, hierarchical association rules)
- techniques for efficiently applying KDD algorithms to large data sets
- overview on further mining techniques and applications

Learning Outcomes

After completing the course, students are able to

- explain the different kinds of knowledge discovery,
- compare and assess concrete method, and
- decide which method can reasonably be used in which situation.

Bibliography

- M. Ester, J. Sander: Knowledge Discovery in Databases: Techniken und Anwendungen. Springer Verlag, 2000.
- M.H. Dunham: Data Mining – Introductory and Advanced Topics, Prentice Hall. 2003.
- P.-N. Tan, M. Steinbach, V. Kumar: Introduction to Data Mining. Addison Wesley. 2006.
- J. Han, M. Kamber: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers. 2000.
- M. Berthold, D.J. Hand (eds.): Intelligent Data Analysis: An Introduction. Springer Verlag, Heidelberg. 1999. - == Module compatibility
- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- active participation at the exercises
- homeworks
- final examination (usually as oral examination)

Responsible persons

Prof. Dr. Stefan Conrad

Cryptocomplexity 2

ECTS credits	Total hours	Contact hours	Self-study hours
10 CP	300 hours	90 hours	210 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (4 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Remarks

- 15 CP for students studying according to PO 2005.

Content

This module provides an advanced understanding of cryptology, introduces some central public-key cryptosystems and digital signature schemes, and discusses their security that often relies on the computational complexity of suitable problems. Therefore, advanced topics of complexity theory are dealt with as well, in particular with the goal to understand and apply methods for the complexity-theoretical classification of important problems. A special focus is given to understanding the close connections between these two fields.

Advanced Cryptology

- Recap: Tasks and Goals of Cryptology, Some Mathematical Foundations, and RSA
- Diffie-Hellman and the Problem of the Discrete Logarithm
- ElGamal's Protocols (the Public-Key Cryptosystem and Digital Signatures of ElGamal and Their Security)
- Rabin's Public-Key Cryptosystem
- Arthur-Merlin Games and Zero Knowledge
- Further Protocols

Advanced Complexity Theory

- Recap: Tasks and Goals of Complexity Theory
- Randomized Algorithms and Complexity Classes (PP, RP, and ZPP: Monte-Carlo and Las-Vegas Algorithms; Quantors and BPP; Graph Isomorphism and the Arthur-Merlin Hierarchy)
- The Boolean Hierarchy (Problems in DP; Structure and Properties of the Boolean Hierarchy over NP; Exact Graph Colorability)
- The Polynomial Hierarchy (Oracle Turing Machines; Structure and Properties of the Polynomial Hierarchy; Complete Problems; the Boolean Hierarchy Collapses the Polynomial Hierarchy)

Learning Outcomes

After completing the course, students are able to

- explain the idea of public-key cryptography,
- unassistedly carry out and explain mathematical (in particular, number-theoretical) calculations that are used in asymmetric encryption and decryption as well as with digital signatures,
- evaluate the security of public-key cryptosystems (RSA, ElGamal, etc.) and digital signatures,
- reason why which of the considered cryptosystems are insecure under which conditions,
- explain possible countermeasures against this,
- explain the motivation and definition of complexity hierarchies,
- describe and evaluate the complexity of natural problems,
- unassistedly design reductions between problems to show their lower bounds and provide a corresponding proof of correctness, and

- assess connections between hierarchies of complexity classes and with cryptological applications.

Bibliography

- Jörg Rothe: Komplexitätstheorie und Kryptologie. Eine Einführung in Kryptokomplexität. eXamen.Press. Springer-Verlag, Heidelberg, 2008. 1. Auflage.
- Jörg Rothe: Complexity Theory and Cryptology. An Introduction to Crypto-complexity. EATCS Texts in Theoretical Computer Science, Springer-Verlag. Heidelberg, 2005. 1. Auflage.
- Further literature is provided during the course.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- active and successful participation in the exercises
- written exam (exam, usually 90 minutes)

Responsible persons

Prof. Dr. Jörg Rothe

Master's Seminar: Current Topics in Theoretical and Computational Biology

ECTS credits	Total hours	Contact hours	Self-study hours
2 CP	60 hours	15 hours	45 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (1 HPW)	every semester	Master Computer Science	English

Content

- Modeling of biological systems
- Bioinformatics algorithms
- Methods for evaluating complex biological data

Learning Outcomes

After completing the course, students are able to

- outline current topics in theoretical and computational biology,
- present scientific content in a lecture, and
- conduct an oral scientific discourse in English.

Bibliography

- Selected publications regarding the topic of the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- Successful presentation of a project
- Active participation in the discussions during the seminar

Responsible persons

Prof. Dr. Martin Lercher

Master's Seminar: Algorithms for Graph Theoretical Concepts in Computer Science

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Tutorial (2 HPW) Seminar (2 HPW)	irregular	Master Computer Science	German

Content

In this seminar, specially selected topics from the field of Algorithms for graph-theoretic concepts in computer science are dealt with. The selection varies from seminar to seminar and is based on the current research topics of the working group.

Learning Outcomes

After completing the course, students are able to

- review and evaluate current topics in the field of graph algorithms
- systematically process scientific publications
- summarize and present content from the literature in a structured way
- analyse and critically reflect scientific literature

Bibliography

- Selected publications regarding the topic of the course.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Contents of modules Algorithms for Graphs I or Algorithms for Graphs II

Conditions for awarding credit points

- written summary of the chosen topic
- adequate presentation of the topic
- active participation in discussions

Responsible persons

Prof Dr. Egon Wanke

Master's Seminar: Algorithms for Perfect Graphs

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Content

Perfect graphs are one of the best studied graph classes. Numerous problems can be solved easier or faster on perfect graphs than on general graphs. In this course we consider different types of perfect graphs, e.g. trees, bipartite graphs, interval graphs, comparability graphs, threshold graphs, and chordal graphs, and some classical problems on these classes.

Learning Outcomes

After completing the course, students are able to

- gain insights into some of the most important topics, results and methods from the field of perfect graphs,
- prepare and give a good talk,
- listen to and critically discuss other students' talks, and
- prepare a good written paper on the talk topic.

Bibliography

- M. Golumbic: Algorithmic Graph Theory and Perfect Graphs. Elsevier. 2004.
- F. Gurski, I. Rothe, J. Rothe, E. Wanke: Exakte Algorithmen für schwere Graphenprobleme. Springer Verlag. 2010.
- J.L. Ramirez-Alfonsin, Bruce A. Reed (Editors): Perfect Graphs. Wiley. 2010.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formally: Bachelor's students must meet the requirements for the anticipation of Master's modules.

Conditions for awarding credit points

- Active participation in the talks and exercises
- handing in the exercises
- passing the written exam at the end of the semester

Responsible persons

Priv.-Doz. Dr. Frank Gurski

Master's Seminar: Operating System Development with Rust

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	every winter semester	Master Computer Science	German

Content

This seminar focuses on research in the area of OS development in Rust.

Rust is a programming language focusing on three things: safety, speed, and concurrency. It accomplishes many of these goals through strong compile-time checks, allowing for very little overhead at runtime. Performance is comparable to C or C++, while being free of many of the problems caused by things like dangling pointers, buffer overflows, and iterator invalidation. This makes Rust interesting for operating system development.

Learning Outcomes

After completing the course, students are able to

- research relevant contents of a given topic and to reproduce them in a written and oral form,
- evaluate the readability and technical quality of other papers,
- plan and hold a presentation,
- ask and answer adequate questions, and
- give and take constructive feedback.

Bibliography

- Relevant literature is provided during the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formally: Bachelor's students must meet the requirements for the anticipation of Master's modules.

Conditions for awarding credit points

- Writing a paper for the given topic
- Contributing to the peer review process
- Presentation of the given topic
- Participate in discussions

Responsible persons

Prof. Dr. Michael Schöttner

Master's Seminar: Combinatorial Optimization in Bioinformatics

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	every summer semester	Master Computer Science	English

Content

Many problems in bioinformatics can be formulated as combinatorial optimization problems. This module teaches advanced concepts to recognize, model and solve these problems. Topics are inspired by recent publications in important areas of bioinformatics including

- genomics, transcriptomics, proteomics and related -omics fields
- systems and network biology
- structural biology
- phylogenetics and clustering
- genotype/phenotype screens
- applications to cancer and other medical research as well as plant science and other areas of biology

Learning Outcomes

After completing the course, students are able to

- extract significant contents of a given topic and to reproduce them in a written and oral form,
- prepare discussion points to enable a critical discussion on a topic,
- evaluate the readability and technical quality of scientific publications,
- plan and hold a presentation and create a written report,
- ask and answer adequate questions and to give and take constructive feedback.

Bibliography

- Selected publications regarding the topic of the course.

Module compatibility

- Elective Area Theoretical Computer Science
- Wahlpflichtbereich Praktische Informatik
- Major Subject
- Individual Supplement
- Elective Area Master study programme Artificial Intelligence and Data Science

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Contents of modules *Algorithms and Data Structures*, *Theoretical Computer Science*, *Mathematics for Computer Science 3* and *Scientific Methods*

Conditions for awarding credit points

- Accepting a topic
- Active participation
- Presentation of a topic and preparation of discussion points
- Written report
- Peer Review

Responsible persons

Prof. Dr. Gunnar W. Klau

Master's Seminar: Computational Multiomics

ECTS credits	Total hours	Contact hours	Self-study hours
2 CP	60 hours	15 hours	45 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	irregular	Master Computer Science	English

Content

Modern laboratory equipment enables high-throughput experiments that provide detailed insights into molecular biology. These give rise to several branches of research including genomics, transcriptomics, proteomics, metabolomics, epigenomics, and metagenomics, with -omics meaning “the total of some sort”. Omics experiments produce massive amounts of data that are subject of a variety of challenging computational problems.

In this class,

- various multiomics experiments,
- methods for multiview learning, multimodal analysis, and deep clustering, and
- applications in multiomics data analysis presented and discussed.

Participants will be assigned individual topics. The selection and matching between topics and participants will be determined in the first meeting of the seminar.

Learning Outcomes

After completing the course, students are able to - outline the current state of research on multiomics data integration, - critically disseminate scientific papers, - give oral presentations on scientific topics, and - write a scientific review.

Bibliography

- Selected publications regarding the topic of the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- Scientific essay on an assigned topic
- Oral presentation

The final grade is composed of 75% written and 25% oral (presentation+participation) performance.

Responsible persons

Jun.-Prof. Dr. Daniel Dörr

Master's Seminar: Databases and Information Systems

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	irregular	Master Computer Science	German und English

Content

In this seminar, basic and current research of the areas “Information Retrieval”, “Natural Language Processing”, “Knowledge Discovery in Databases”, and “Machine Learning” is considered building upon the methods presented in one of the underlying lectures. Thereby, improvements of known methods as well as new methods can be considered.

Learning Outcomes

After completing the course, students are able to

- comprehend and assess current methods,
- systematically process scientific publications,
- summarize and present content from the literature in a structured way, and
- analyse and critically reflect scientific literature.

Bibliography

- None

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Contents of one of the following modules: *Knowledge Discovery in Databases*, *Knowledge Discovery in Databases - Selected Topics*, *Multimedia Database Systems*, *Natural Language Processing* (depending on the thematic focus of the seminar, which will be published in HIS/LSF)

Conditions for awarding credit points

- written summary of the chosen topic
- adequate presentation of the topic
- active participation in discussions

Responsible persons

Prof. Dr. Stefan Conrad

{Master's Seminar: Deep Learning

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	irregular	Master Computer Science	German

Remarks

- For didactic reasons, a maximum of 25 students can participate in the seminar.

Content

In the seminar, the contents of the lecture [module] **Machine Learning** are to be further deepened and expanded, particularly in the area of Deep Learning. Therefore, various topics from the field of Deep Learning will be presented and discussed in this seminar. These topics include, for example, Deep Convolutional Networks, Deep Autoencoders, Generative Adversarial Networks, etc.

Each week, one or two topics will be presented and discussed by the participants in the form of a presentation. This involves not only the content but also how to deliver a good scientific presentation. In addition, the participants will create a brief written report of their presentation.

Learning Outcomes

After completing the course, students are able to

- Explain methods and techniques from the field of Deep Learning,
- give a presentation on a topic from this field, and
- select appropriate methods for specific questions and problems.

Bibliography

- Relevant literature is provided during the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Contents of modules *Machine Learning* und *Mathematics for Computer Science 1–3* (alternatively *Calculus I* and *Linear Algebra I*)

Conditions for awarding credit points

- Written report on the selected topic (2/3 of the grade)
- Presentation of one's own topic (1/3 of the grade)
- Active participation in discussions

Responsible persons

Prof. Dr. Dominik Heider

Master's Seminar: Digital Innovation and Entrepreneurship

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	irregular	Master Computer Science	German/English

Content

The seminar covers a selection of current research topics in the area of Digital Innovation and Entrepreneurship.

- Students learn theories, concepts, processes, tools, and methods surrounding IT-based innovation, business models, user experience, and entrepreneurship.
- Topics will be introduced and assigned in the first session.
- During the semester, students will work on their seminar paper.
- The results will be presented and discussed in the middle and at the end of the semester.
- In addition, an introduction to scientific writing is given at the beginning.

Learning Outcomes

After completing the course, students are able to

- independently prepare and interpret the most important theories, concepts, methods and results related to the field of digital innovation and entrepreneurship,
- analyze scientific literature in order to independently define relevant research problems,
- apply conceptual, quantitative, or qualitative research methods to analyze these problems,
- write and review seminar papers,
- critically reflect, present, and discuss the topic, research design, and results in the context of the seminar,
- provide appropriate feedback on complex challenges, and
- develop teamwork skills through collaboration with fellow students.

Bibliography

- Selected publications regarding the topic of the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules. Preference will be given to Master's students when space is allocated.

Conditions for awarding credit points

- Written elaboration of the topic to be worked on (if necessary in groups)
- Presentation of the topic (if necessary in groups)
- Active participation in discussions

Responsible persons

Prof. Dr. Steffi Haag

Master's-Seminar: Collective Decision-Making

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	irregular	Master Computer Science	German

Content

In this master-seminar, talks are given about selected topics from the following areas that are also offered in the modules of theoretical computer science with lectures and exercises and that each fall under the broader term of collective decision-making: Computational Social Choice, Algorithmic Game Theory, and Fair Division. This triad of "voting, playing, dividing" is complemented by the related topics Judgment Aggregation, Participatory Budgeting and Argumentation Theory. For all talk topics, the focus is on the algorithmic and complexity-theoretic properties of the related problems.

- Computational Social Choice
- Algorithmic Game Theory
- Fair Division
- Judgment Aggregation
- Participatory Budgeting
- Argumentation Theory

Learning Outcomes

After completing the course, students are able to

- unassistedly create a written manuscript about a scientific topic from the original literature,
- present a talk about this special topic and create clearly structured and designed slides to this end,
- provide correct responses to questions about their topic,
- assess the quality of another student's manuscript in a double-blind peer-reviewing critically but also fairly, and
- discuss the talks of other students critically and fairly and pose discerning questions about them.

Bibliography

Originalliteratur, die zum Beispiel in den folgenden Büchern zitiert wird:

- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division. Springer Texts in Business and Economics, Springer-Verlag. Heidelberg, 2015. 1. Auflage.
- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen. Spektrum Akademischer Verlag (Springer). Heidelberg, 2011. 1. Auflage.
- Further literature is provided during the course.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- written manuscript about the assigned topic
- presentation of a talk
- double-blind peer-review of another student's manuscript

Remark: This modul is graded according to previously known criteria. Specifically, the grade results from the evaluation of the quality of the manuscript and of the talk, both with equal weight, provided they both are evaluated as "passed". In addition, the grade can be improved by bonus performances, such as a helpful review or an active participation in the discussion of the talks of other students.

Responsible persons

Prof. Dr. Jörg Rothe

Master's-Seminar: Complexity and Cryptology

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	irregular	Master Computer Science	German

Content

In this master-seminar, talks are given about selected topics from computational complexity theory and cryptology. For the complexity-theoretic talks, the focus often (but not always) is on problems from Computational Social Choice, Algorithmic Game Theory, Fair Division, Judgment Aggregation, Participatory Budgeting, and Argumentation Theory. For the cryptologic talks, the main focus is on modern public-key cryptology.

Learning Outcomes

After completing the course, students are able to

- unassistedly create a written manuscript about a scientific topic from the original literature,
- present a talk about this special topic and create clearly structured and designed slides to this end,
- provide correct responses to questions about their topic,
- assess the quality of another student's manuscript in a *double-blind peer-reviewing* critically but also fairly, and
- discuss the talks of other students critically and fairly and pose discerning questions about them.

Bibliography

Originalliteratur, die zum Beispiel in den folgenden Büchern zitiert wird:

- Jörg Rothe: Komplexitätstheorie und Kryptologie. Eine Einführung in Kryptokomplexität. eXamen.Press. Springer-Verlag, Heidelberg, 2008. 1. Auflage.
- Jörg Rothe: Complexity Theory and Cryptology. An Introduction to Crypto-complexity. EATCS Texts in Theoretical Computer Science, Springer-Verlag. Heidelberg, 2005. 1. Auflage.
- Jörg Rothe (ed.): Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division. Springer Texts in Business and Economics, Springer-Verlag. Heidelberg, 2015. 1. Auflage.
- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen. Spektrum Akademischer Verlag (Springer). Heidelberg, 2011. 1. Auflage.
- Further literature is provided during the course.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- written manuscript about the assigned topic
- presentation of a talk
- double-blind peer-review of another student's manuscript

Remark: This modul is graded according to previously known criteria. Specifically, the grade results from the evaluation of the quality of the manuscript and of the talk, both with equal weight, provided they both are evaluated as "passed". In addition, the grade can be improved by bonus performances, such as a helpful review or an active participation in the discussion of the talks of other students.

Responsible persons

Prof. Dr. Jörg Rothe

Master's Seminar on NP-hard problems

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Content

The hardness of some problems gives reason to find creative methods to approach the solution of a problem. For example, one can solve sub-problems or examine the problem for restricted classes, to find approximation algorithms or parameterized algorithms, or to solve the problem efficiently under a given constraint. In this module several of these methods are investigated. The goal of this module is to provide students with insights into some important or current research topics, results and methods in the field of NP-hard problems.

Learning Outcomes

After completing the course, students are able to

- get insights into some current research findings themselves and explain and present results from research findings,
- explain and formally define the hard problems discussed,
- explain the algorithms and methods discussed and apply them to specific inputs,
- perform run-time analyses of the algorithms,
- set tasks for themselves and
- produce material sufficiently good for other students to solve these tasks.

Bibliography

- Selected publications regarding the topic of the course.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formally: Bachelor's students must meet the requirements for the anticipation of Master's modules.

Conditions for awarding credit points

- Active participation in the talks and exercises
- handing in the exercises
- passing the written exam at the end of the semester

Responsible persons

Priv.-Doz. Dr. Frank Gurski

Master's Seminar: Property Testing

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	irregular	Master Computer Science	English

Content

We study algorithmic techniques that allow us to solve relaxed decision problems efficiently such as testing whether a sparse graph is connected or „far away“ from being connected in sublinear time. By sampling, we can obtain partial information of an input locally, which provides structural information about the global input. Individual topics usually deal with a specific problem setting, a specific testing algorithm, and a running time and accuracy analysis thereof.

- function properties (e.g., monotonicity, linearity)
- graph properties (e.g., connectivity in bounded-degree graphs, bipartiteness in dense graphs)

Learning Outcomes

After completing the course, students are able to

- extract significant contents of a given topic and to reproduce them in a written and oral form,
- evaluate the readability and technical quality of other abstracts,
- plan and hold a presentation and
- ask and answer adequate questions and to give and take constructive feedback.

Bibliography

- Oded Goldreich: Introduction to Property Testing. Cambridge University Press. 2017. 1st Edition
- Dana Ron: Algorithmic and Analysis Techniques in Property Testing. Foundations and Trends in Theoretical Computer Science. 2010. 5(2):73–205

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics
- Elective Area Master study programme Artificial Intelligence and Data Science

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Contents of modules *Algorithms and Data Structures*, *Theoretical Computer Science*, *Mathematics for Computer Science 3* and *Scientific Methods*

Conditions for awarding credit points

- written abstract on a given topic
- peer review

- presentation of the topic
- asking and answering questions

Responsible persons

Dr. Anja Rey

Master's Seminar: Reproducibility in Bioinformatics Research

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	jedes Wintersemester	Master Computer Science	English

Content

Scientific data analysis is becoming increasingly important in research. The combination of huge data volumes, lack of standards and incentives, poor programming practices, and inadequately documented parameters and versions has led to a reproducibility crisis. Scientific software is often inadequately documented, inefficiently programmed, and not well-maintained. In this seminar, we will confront these issues by attempting to reproduce the data analysis of selected original papers from the area of bioinformatics. In addition, we will learn how to improve the analysis steps, e.g., by using workflow management tools such as Snakemake and depositing the pipelines on certified online storage services.

Students organize themselves in small groups and work together on a topic. They present the research and work together on its reproducibility, typically using a shared git repository. They present their results and write a report about the original research and their reproducibility efforts.

Learning Outcomes

After completing the course, students are able to

- explain the importance of reproducibility in science,
- formulate the pitfalls of reproducibility and know about the reproducibility crisis,
- apply techniques to ensure reproducibility in computational science,
- extract significant contents of a given topic and to reproduce them in a written and oral form,
- prepare discussion points to enable a critical discussion on a topic,
- evaluate the readability and technical quality of scientific publications,
- plan and hold presentations and create a written report, and
- ask and answer adequate questions and to give and take constructive feedback.

Students also have demonstrated to work as a group and to divide the work in a fair manner.

Bibliography

- Selected publications regarding the topic of the course.

Module compatibility

- Wahlpflichtbereich Praktische Informatik
- Major Subject
- Individual Supplement
- Elective Area Master study programme Artificial Intelligence and Data Science

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

- Contentual: Contents of modules *Algorithms and Data Structures*, *Theoretical Computer Science*, *Mathematics for Computer Science 3* and *Scientific Methods*

Conditions for awarding credit points

- Accepting a topic
- Active participation
- Presentation of a topic and preparation of discussion points
- Written report
- Peer Review

Responsible persons

Prof. Dr. Gunnar W. Klau

Master's Seminar: User Experience (UX) Design

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	irregular	Master Computer Science	English

Content

The seminar covers current research topics around user experience (UX) design, which integrates the disciplines of computer science, esp. human-computer interaction (HCI), psychology, and business. UX design is concerned with the analysis, creation, and optimization of user experiences.

- Students learn theories, concepts, processes, tools, and methods for exploring users' cognitions, emotions, and behaviors in interaction with digital technologies.
- Topics will be introduced and assigned in the first session.
- During the semester, students will work on their seminar paper.
- The results will be presented and discussed in the middle and at the end of the semester.
- In addition, an introduction to scientific writing is given at the beginning.

Learning Outcomes

After completing the course, students are able to

- independently prepare and interpret the most important theories, concepts, methods and results related to the field of user experience (UX) design,
- analyze scientific literature in order to independently define relevant research problems,
- apply conceptual, quantitative, or qualitative research methods to analyze these problems,
- write and review seminar papers,
- critically reflect, present, and discuss the topic, research design, and results in the context of the seminar,
- provide appropriate feedback on complex challenges, and
- develop teamwork skills through collaboration with fellow students.

Bibliography

- Selected publications regarding the topic of the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules. Preference will be given to Master's students when space is allocated.

Conditions for awarding credit points

- Written elaboration of the topic to be worked on (if necessary in groups)
- Presentation of the topic (if necessary in groups)
- Active participation in discussions

Responsible persons

Prof. Dr. Steffi Haag

Master's Seminar: Scientific Methods

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	every winter semester	Master Computer Science	German

Content

The students gain insights into essential scientific methods and give a lecture in the seminar on a special topic from the original literature. The presentations of the other students are critically discussed among all participants. Each lecture includes a written elaboration and there is a mutual assessment of the written elaborations of the students among themselves in a double-blind peer-reviewing procedure. The anonymity of both the author and the reviewer is preserved.

- How to write a scientific article?
- How do you write a literature review?
- Scientific reasoning with diagrams
- Scientific creativity: role of questions, interdisciplinarity, analogies and anthropomorphisms
- Unbiased data analysis
- Relationship between data and hypotheses
- Confirmation bias and contradictions

Learning Outcomes

After completing the course, students are able to

- describe essential scientific methods,
- prepare a scientific paper,
- prepare and give a good presentation,
- critically discuss presentations by other students and the content presented,
- summarize, communicate and defend subject-related positions,
- evaluate scientific elaborations of other participants critically and fairly.

Bibliography

- Selected publications regarding the topic of the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- Written elaboration on the chosen topic
- Giving a lecture
- Evaluation of two other written elaborations
- All student contributions to the module are included in equal parts in the final grade, i.e. written elaboration, two assessments of elaborations by fellow students, seminar presentation.

Responsible persons

Prof. Dr. Lercher, Dr. Mayo Röttger

Methods for Population Genetics

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	45 hours	105 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (1 HPW)	every winter semester	Master Computer Science	English

Content

Population Genetics studies the distribution of genetic information across populations under varying conditions. This lecture provides the necessary methods to answer questions such as: Is there evidence for selection pressure on a certain gene? How can ancient migration patterns be retraced based on genetic information? How can quantitative traits such as disease susceptibility be attributed to genetic loci?

- Hardy-Weinberg principle
- Genetic drift and Wright-Fisher model
- Kolmogorov forward and backward equations
- Mutation and selection
- Linkage disequilibrium
- Population structure and inbreeding
- Haplotype phasing and imputation
- Genome-wide association studies and quantitative traits.

Learning Outcomes

After completing the course, students are able to

- explain the fundamental principles and terms used in population genetics,
- analyze and compare the effects of mutation, migration and selection on genotypes in a population,
- describe analyses of genetic variation in the context of phenotypic traits, and
- apply the algorithmic and statistical concepts of population genetics to modern sequencing data sets.

Bibliography

- Hartl and Clark: Principles of Population Genetics. Sinauer. 2007.
- John Gillespie: Population Genetics - A Concise Guide. Johns Hopkins University Press. 2004.
- Matthew Hamilton: Population Genetics. Wiley-Blackwell. 2009.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Elective Area Master study programme Artificial Intelligence and Data Science
- Application subject for Minor area Master study programme Mathematics
- Individual Supplement

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- To be eligible to take the exam, students have to earn 50% of the points from the assignment sheets.
- Oral exam (usually 20 minutes)

Responsible persons

Prof. Dr. Tobias Marschall

Model Checking

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	every summer semester	Master Computer Science	German

Content

In this module we study automatic verification of hardware and software systems using model checking algorithms and tools.

- Model checking algorithms for the verification of temporal (safety and liveness) and regular properties
- Principles and practical use of temporal logics (LTL, CTL, CTL*, ...)
- Use of model checking tools (e.g., SMV, Spin, ProB, ...)
- Principles of advanced methods (e.g., binary decision diagrams, partial order reduction, symmetry, ...)

Learning Outcomes

After completing the course, students are able to

- present, apply and compare different techniques of verification,
- develop simple system specifications and check them automatically using model checking tools,
- formalise temporal and regular properties using an adequate formalism and
- develop their own model checking tool.

Bibliography

- Christel Baier und Joost-Pieter Katoen. Principles of Model Checking. MIT Press, 2008.
- M. Clarke, O. Grumberg, D.A. Peled. Model Checking, Second Edition. MIT Press. 2018.

Module compatibility

- Elective Area Theoretical Computer Science
- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- seminar presentation and successful completion of a model checking project
- oral (20-30 minutes) or written (90 minutes) exam, depending on the number of students

Responsible persons

Prof. Dr. Michael Leuschel

Multimedia Database Systems

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Remarks

- 7.5 CP for students studying according to PO 2005.

Content

This module covers basic notions and methods for dealing with multimedia objects in retrieval systems using image data as example:

- basic notions of information retrieval (definition; evaluation measures)
- multimedia information retrieval (with focus on image databases)
- feature extraction, selection and transformation
- distance functions; curse of the dimensionality
- efficient algorithms and data structures for similarity search in high-dimensional feature spaces (multi-dimensional index structures, similarity search)

Learning Outcomes

After completing the course, students are able to

- explain the basic tasks in multimedia information retrieval,
- select or design adequate distance functions, and
- design an own multimedia database system.

Bibliography

- I. Schmitt: Ähnlichkeitssuche in Multimedia-Datenbanken. Oldenbourg-Verlag, München. 2005.
- K. Meyer-Wegener: Multimedia-Datenbanken – Einsatz von Datenbanktechnik in Multimedia-Systemen. Teubner, Stuttgart. 2003. 2. Auflage
- B. Prabhakaran: Multimedia Database Management Systems. Kluwer Academic Publishers, Massachusetts. 1997.
- R. Veltkamp, H. Burkhardt, H.-P. Kriegel (eds.): State-of-the-Art in Content-Based Image and Video Retrieval. Kluwer Academic Publishers. 2001.
- R. Băza-Yates, B. Ribeiro-Neto: Modern Information Retrieval. Pearson / Addison Wesley. 1999.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- active participation in the exercises
- homeworks
- final examination (usually as oral examination)

Responsible persons

Prof. Dr. Stefan Conrad

Natural Language Processing

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	every second winter semester	Master Computer Science	English

Remarks

- 7.5 CP for students studying according to PO 2005.

Content

- introduction to Natural Language Processing (NLP)
- NLP pipeline and methods (language identification, segmentation, POS tagger, stemming and lemmatization, stop word elimination, dependencies, thesaurus, ...)
- POS tagging in detail (rule-based; hidden markov model, ...)
- distributional semantics (vector representations, word embeddings, deep learning architectures for NLP)
- annotation process (guidelines, gold standard, measurements for inter-annotator agreement)

Learning Outcomes

After completing the course, students are able to

- explain basic methods and algorithms for NLP,
- design a NLP pipeline for a dedicated task and implement it using adequate libraries,
- know how to evaluate NLP algorithms and whole pipelines, and
- interpret the results of such evaluations.

Bibliography

- C. Manning, H. Schütze: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge. 1999.
- Further literature is provided during the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- active participation in the exercises
- homeworks

- final examination (usually as oral examination)

Responsible persons

Prof. Dr. Stefan Conrad

Practical: Implementing Transformer Models

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Lab (2 HPW)	irregular	Master Computer Science	English

Content

- Detailed study and implementation of the Transformer model, building an intuitive understanding of its unique architecture and the attention mechanism.
- Practical application of computational aspects of the Transformer model, including the scaling of dot products and shared parameter mechanisms, such as embedding vectors.
- Training a Transformer model tailored for machine translation, including the preparation and pre-processing of a translation dataset.
- Exploration of GPU utilisation, parallel training strategies and effective resource allocation for machine learning training.

Learning Outcomes

By the end of the module, students will be able to:

- interpret the architectural nuances and computational aspects of Transformer models, including the attention mechanism and multi-head attention layers.
- adopt scientific research skills, such as setting up independent projects, implementing models based on academic literature, and producing scientifically sound, reproducible results.
- apply knowledge of optimal GPU utilisation, including the setup and management of parallel training environments and effective resource allocation for machine learning training.

Bibliography

- Attention is all you need. (<https://arxiv.org/pdf/1706.03762.pdf>)

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor students must meet the requirements for the anticipation of Master modules.
- Contentual: Content of the *Machine Learning* module

Conditions for awarding credit points

Active participation in the practical. Passing grades for presentation and report.

Responsible persons

Prof. Dr. Milica Gašić, Carel van Niekerk

Representations in Natural Language Processing

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	English

Content

Word embeddings model natural language by associating to each word in the vocabulary a vector in an ambient space, such that words which have similar meaning are close together. When data is encoded as points in a high-dimensional space, this is called an embedding.

Words which are used in similar contexts usually have similar meanings (this is also known as the distributional hypothesis). Thus it is possible to acquire meaningful word representations from unlabelled data, such as from Wikipedia paragraphs, books, or news headlines.

In this lecture, we study recently published techniques for building both static and contextual embeddings. In a static embedding, there is one fixed vector for each word in the vocabulary. In a contextual embedding, the vector we associate to a word is distinct if it is used in a different context. For example, the word 'bank' has separate representations depending on whether it refers to the side of a river, or to a financial institution.

Another important but challenging aspect is the evaluation of the quality of a word embedding, which form the basis of many natural language processing tasks such as document search, classification, information retrieval, language translation and sentiment analysis.

Overview:

- Static word embeddings: Frequency based methods, word2vec, GloVe, fastText
- Contextual word embeddings: Recurrent methods, Language Models, ELMo, Transformers, BERT and Masked Language Models, Contrastive Learning, Sentence Embeddings
- Intrinsic evaluation of word embeddings and applications in dialog systems
- Bias in word embeddings, Sentiment
- Multi-lingual word embeddings
- Hyperbolic word embeddings, Topological Data Analysis and its applications

Learning Outcomes

After completing the course, students are able to

- compare methods for constructing word embeddings,
- implement and apply the underlying algorithms,
- name applications for word embeddings in natural language processing and dialog systems, and
- independently review and evaluate scientific publications.

Bibliography

- Selected publications regarding the topic of the course.
- D. Jurafsky, J. Martin: Speech and Language Processing. Prentice Hall. Upper Saddle River, NJ, 2008. 2nd edition.
- J. Eisenstein: Introduction to Natural Language Processing. MIT Press. Cambridge, MA, 2019. 1st edition.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor students must meet the requirements for the anticipation of Master modules.
- Formal: Students who have completed the master's seminar on word embedding are not allowed to participate in this course.
- Contentual: Content of the *Machine Learning* module

Conditions for awarding credit points

- Active and successful participation in exercises
- Passing the exam (homework)

Responsible persons

Prof. Dr. Milica Gašić, Benjamin Matthias Ruppik

Safety-Critical Systems

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (1 HPW) Lab (1 HPW)	every summer semester	Master Computer Science	German

Remarks

- 7.5 CP for students studying according to PO 2005.

Content

This course covers the development of safety critical systems using the formal B-method. The B-method was invented by Jean-Raymond Abrial and has good tool support (AtelierB, Rodin, ProB).

We cover these themes:

- Overview of formal methods in general, along with their practical use in systems development and within industrial norms
- Foundations of the B language
- Invariants and proof obligations
- Refinement
- Overview of industrial B models
- Other correctness criteria: well-definedness and temporal LTL properties

Learning Outcomes

After completing the course, students are able to

- compare the foundations of the B language (syntax and semantics) with classical programming languages,
- check the type-correctness of B expression, as seen in the exercises,
- express queries as B expressions, as seen in the exercises,
- develop formal B specifications, thereby using existing B tools (mainly Rodin and ProB),
- check formal specifications for invariant violations or refinement errors,
- prove simple specifications correct both by hand and using an automated proof environment (Rodin) and
- use refinement to structure complex systems and proofs.

Bibliography

- J.R. Abrial: Modeling in Event-B: System and Software Engineering. Cambridge University Press. 2010. ISBN 0521895561.

Module compatibility

- Elective Area Theoretical Computer Science
- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- successful completion of mandatory exercises
- written exam

Responsible persons

Prof. Dr. Michael Leuschel

Stochastic Models of Biological Systems

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	every semester	Master Computer Science	English

Content

- Cell birth-and-death, moment-generating functions.
- Diffusion models and Brownian dynamics, comparison of deterministic versus stochastic approaches.
- Protein production, the Totally Asymmetric Simple Exclusion Process, its analytical solution in mean field.
- Cytoskeletal transport, directed transport properties of three dimensional structures.
- Polysaccharide synthesis, complex enzymatic processes.

The course describes the stochastic modelling (analytical and numerical) of the biological systems from cell biology above listed. Key algorithms used throughout the module are Monte Carlo and Gillespie, implemented in C. The module is highly interdisciplinary and includes knowledge and methods from Mathematics, Biology, Physics, and Chemistry, with a strong focus on algorithms and programming in the tutorials. The methods and tools learnt during this course also apply to other fields than Biology. Specifically, they are used in theoretical Physics of both dilute and condensed systems, and Engineering. The programming language C is widely spread in these fields since it is well suited for computer intensive numerical methods.

Learning Outcomes

After completing the course, students are able to

- distinguish and recognise key biological features that justify the use of the models studied,
- explain modelling approaches based on the underlying biology,
- develop and implement stochastic algorithms using the programming language C++,
- simulate the dynamics of the systems studied and critically evaluate the quality of their results,
- reproduce all analytical calculations executed in the course,
- explain physico-chemical parameters that influence the dynamics of the processes studied.

Bibliography

- None

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- At least 50 percent of the points from the exercises
- Passing final exam (written exam, usually 90 minutes)

Responsible persons

Dr. Adélaïde Raguin, Prof. Dr. Martin Lercher

System Software for Big Data Computing

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	every summer semester	Master Computer Science	German

Content

- big-data systems: batch-, in-memory, stream-, Graph-Processing; distributed machine learning, ...
- cloud systems: storage solutions, virtualization, messaging and coordination services,
- operating systems for manycore, NVRAM (persistent memory), RDMA (Remote Direct Memory Access), GPUs, ...

Learning Outcomes

After completing the course, students are able to

- research relevant contents of a given topic and to reproduce them in a written and oral form
- evaluate the readability and technical quality of other papers
- plan and hold a presentation
- ask and answer adequate questions and
- give and take constructive feedback.

Bibliography

- Relevant literature is provided during the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formally: Bachelor students must meet the requirements for the anticipation of Master modules.

Conditions for awarding credit points

- Writing a paper for the given topic
- Contributing to the peer review process
- Presentation of the given topic
- Participate in discussions

Responsible persons

Prof. Dr. Michael Schöttner

Transaction Management

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Remarks

- 7.5 CP for students studying according to PO 2005.

Content

This module deals with the basics of transaction management. For thatm formal models are introduced, their properties are analysed and their implementation in real systems is discussed. In detail the following topics are covered:

- transaction properties und requirements (ACID, ...)
- serializability (final-state, view, conflict serializability and their formal properties)
- properties for abortion (recoverability, avoiding cascading aborts, strictness)
- locking methods and protocols (2 phase locking, tree protocol, non-locking methods)
- recovery (undo/redo, restart, ...)
- structured (nested) transaction models
- formal description of transaction models (ACTA, transaction closures)
- transactions in distributed systems (distributed commit, 2PC protocol)

Learning Outcomes

After completing the course, students are able to

- explain the problems of transaction management in database systems,
- determine the properties of transaction schedules,
- compare and assess recovery methods, and
- asses the possible applications of structures transaction models.

Bibliography

- P. Bernstein, V. Hadzilacos, N. Goodman: Concurrency Control and Recovery in Database Systems. Addison-Wesley, Reading, MA. 1987.
- J. Gray, A. Reuter: Transaction Processing: Concepts and Techniques. Morgan Kaufmann Publishers, San Mateo, CA. 1993.
- G. Weikum, G. Vossen: Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery. Morgan Kaufmann Publishers. 2001.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Contents of module *Databases: An Introduction*

Conditions for awarding credit points

- active participation in the exercises
- homeworks
- final examination (usually as oral examination)

Responsible persons

Prof. Dr. Stefan Conrad

User Experience (UX) Design and Management

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Lab (2 HPW)	every winter semester	Master Computer Science	English

Content

Students explore the user perspective on interacting with digital technologies, thereby becoming more aware of users' needs during product development. Designing and maintaining great user experience (UX) is the best way for both startups and established companies to build trust, retention, and loyalty of staff and customers alike.

The lecture

- teaches the key concepts, methods, and approaches that help design, measure, and manage total UX across organizations and drive value propositions of digital business models.
- discusses established and new methods of UX research for (further) developing digital technologies.
- introduces frameworks to build and lead teams of UX researchers, designers, engineers, product managers.
- employs case studies to transfer and discuss the application of UX design, research, and management in practice.

In the practice sessions, (groups of) students

- practically apply UX research methods and tools (e.g., user interviews, A/B testing, or emotion detection) to investigate users' experiences in interaction with state-of-the-art digital technology prototypes and to deduce implications for product and organizational strategy, development, and design.
- present the results towards peers and experts from research and industry.

Learning Outcomes

After completing the course, students are able to

- define, discuss, and apply the concepts, methods, and tools of analyzing and managing the experiences users perceive in interaction with new digital technologies of startups/organizations,
- measure and analyze user experiences of novel technologies and infer recommendations for technology and policy design and development,
- assess and reflect the social and ethical implications of designing, evaluating, and implementing digital technologies,
- present user research results towards peers, and
- develop skills in collaborative interaction with peers.

Bibliography

- Relevant literature is provided during the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject

- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- Project report in groups
- Presentations in groups
- Active class participation in discussions

Responsible persons

Prof. Dr. Steffi Haag

Advanced Compiler Construction

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Content

The module “Advanced Compiler Construction” deals with advanced topics that are not covered in the module Compiler Construction. Topics of the module are: generation and use of SSA intermediate code, loop analysis, dominator trees, interprocedural analysis, alias analysis, code generation, compiler verification and just-in-time compilation. The students apply acquired knowledge in a programming project. In addition, the participants themselves present a topic from the field of compiler construction in a lecture.

Learning Outcomes

After completing the course, students are able to

- identify advanced compiler construction concepts and problems,
- evaluate when and for which areas of application certain algorithms and methods are advantageous,
- independently create and test an advanced compiler, and
- give a presentation on an advanced topic in the field of compiler construction.

Bibliography

- Andrew W. Appel: Modern Compiler Implementation in Java. Cambridge University Press. 2nd Edition
- Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman: Compilers: Principles, Techniques, and Tools. Addison Wesley. 2nd Edition

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor’s students must meet the requirements for anticipating Master’s modules.
- Contentual: Contents of module Compiler Construction

Conditions for awarding credit points

- Successful completion of compulsory exercises
- Successful development and extension of an own compiler
- Successful presentation on a topic from the field of compiler construction
- Passing the exam

Responsible persons

Dr. John Witulski

Advanced Functional Programmierung: Clojure

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture oder Seminar (2 HPW) Tutorial (2 HPW)	every summer semester	Master Computer Science	German

Content

This module consists of advanced concepts related to Clojure. The following topics are covered:

- generative testing using `test.check`
- optional typing using `clojure.spec`
- transducers
- asynchronous programming using `core.async`
- Datomic-like data bases
- Clojurescript
- depending on interest: further domain-specific libraries and tools

Learning Outcomes

After completing the course, students are able to

- explain and apply complex theoretical concepts of functional programming,
- create web applications using Clojure and Clojurescript and
- create and test large-scale functional programs on their own and in groups.

Bibliography

- Abelson, Sussman: Structure and Interpretation of Computer Programs. MIT Press. 1996. 2nd edition.
- Baldrige: Communicating sequential processes with `core.async`. Infinite Skills. 2015. 1st edition.
- Fogus, Houser: The Joy of Clojure. Manning. 2014. 2nd edition.
- Emerick, Carper, Grand: Programming Clojure. O'Reilly. 2012. 1st edition.
- Rathore, Avila: Clojure in Action. Manning. 2016. 2nd edition.
- Higginbotham: Clojure for the Brave and True. No Starch Press. 2015. 1st edition.
- Miller, Hallway, Bedra: Programming Clojure. Pragmatic Programmers. 2018. 3rd edition.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.
- Contentual: Contents of module *Introduction to Functional Programming*

Conditions for awarding credit points

- Depending on the course implementation: active participation in the seminar sessions
- Depending on the number of participants:
- preferably oral exam (usually 15-30 minutes),
- written exam (exam, usually 90 minutes).

Responsible persons

Philipp Körner, Dr. Jens Bendisposto, Prof. Dr. Michael Leuschel

Advanced Topics in Logic Programming

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (1 HPW) Lab (1 HPW)	every summer semester	Master Computer Science	German

Remarks

- 7.5 CP for students studying according to PO 2005.

Content

In this module we study advanced topics of logic programming and its application to program analysis, verification and artificial intelligence. The modul thus covers constraint logic programming but also the development of interpreters and tools for program analysis and verification in Prolog.

- Advanced Prolog programming techniques
- metaprogramming in Prolog and its applications
- use of constraint logic programming
- implementation of constraint solvers
- development of interpreters in Prolog illustrating concepts such as denotational and operational semantics
- program analysis with abstract interpretation or data flow analysis
- partial evaluation and automatic program optimisation

Learning Outcomes

After completing the course, students are able to

- solve problems with constraint solvers,
- write their own constraint solver,
- encode inference rules for operational semantics in Prolog
- develop an interpreter for a new programming language,
- develop a self-interpreter for Prolog in Prolog,
- describe and compare the foundations of abstract interpretation and its application for program verification,
- develop analysis and verification tools in Prolog,
- apply and write a partial evaluator.

Bibliography

- Lecture Notes

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement

Prerequisites

- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- successful completion of mandatory exercises
- written exam

Responsible persons

Prof. Dr. Michael Leuschel

Courses from other programs or institutes

Some courses from other programs and/or institutes may also be taken by students in the master's program in computer science. These courses can be used in the elective areas *Practical and Technical Computer Science*, *Theoretical Computer Science*, *Individual Specialization*, and *Focus*. The column *Area* indicates whether the course can be used for the area *Theoretical Computer Science* (TH) or *Practical and Technical Computer Science* (PR); an assignment to the respective other area is not possible.

Please be aware of the contentual prerequisites of the respective modules. These prerequisites can be found in the module handbook of the respective study program. If a module is listed here, this does *not* automatically mean that the module can be meaningfully completed with the previous knowledge from the Bachelor of Computer Science alone.

Master AI and Data Science

The modules listed here can be found in the [Module Handbook of the Master AI and Data Science program](#). Please be aware of the contentual prerequisites of the respective modules given in the module handbook. If a module is listed here, this does *not* automatically mean that the module can be completed with the previous knowledge from the Bachelor of Computer Science alone.

Title	Responsible Persons	Cycle	ECTS	Lang.	Area
Applications of Transformer Networks in Bio- and Cheminformatic	Dr. A. Kroll	irregular	5 CP	EN	PR
Computational Argumentation	Jun. Prof. Dr. G. Lapesa	irregular	5 CP	EN	PR
Causality	Prof. Dr. D. Heiders	irregular	5 LP	EN	TH
Clinical Decision Support Systems (CDSS)	Prof. Dr. D. Heider	irregular	5 CP	EN	PR
Computer Vision	Prof. Dr. P. Swoboda	every summer semester	5 CP	EN	PR
Data & Knowledge Engineering (DKE)	Prof. Dr. S. Dietze	irregular	5 CP	EN	PR
Deep Learning	Prof. Dr. T. Dickscheid Prof. Dr. M. Kollmann	every winter semester	5 CP	EN	PR
Deep Learning in Life Science: Generative Models	Prof. Dr. M. Kollmann	every winter semester	5 CP	EN	PR
Deep Learning in Life Science: Representation Learning	Prof. Dr. M. Kollmann	every summer semester	5 CP	EN	PR
Master's Seminar: Computational Argumentation	Jun. Prof. Dr. G. Lapesa	irregular	5 CP	EN	PR
Master's Seminar on Ethics in NLP	Jun. Prof. Dr. G. Lapesa	irregular	5 CP	EN	PR
Information Theory	Dr. P. Arndt	irregular	5 CP	EN	TH

Master's Seminar Advances in Data Science	Prof. Dr. S. Dietze	irregular	5 CP	EN	PR
Master's Seminar on Limits of Computation	A. Lahiri	irregular	5 CP	EN	TH
Master's Seminar on Word Embedding Spaces	Dr. B. Ruppik Prof. Dr. M. Gasic	irregular	5 CP	EN	PR
Neuroimaging and Precision Medicine	Prof. Dr. S. B. Eickhoff Prof. Dr. S. Caspers PD Dr. S. Weis	every winter semester	5 CP	EN	PR
Probabilistic Machine Learning	Prof. Dr. P. Swoboda	irregular	10 CP	EN	PR
Reinforcement Learning	Prof. Dr. S. Harmeling	irregular	5 CP	EN	PR
Relational Databases and Data Analysis	Prof. Dr. S. Conrad	irregular	5 CP	EN	PR
Spectral Graph Theory and Graph Signal Processing	Dr. P. Arndt	irregular	5 CP	EN	PR
Spoken Dialogue Systems	Prof. Dr. M. Gasic	irregular	10 CP	EN	PR
Unsupervised Deep Learning with Applications in Image Analysis	Prof. Dr. T. Dickscheid Prof. Dr. M. Kollmann	irregular	5 CP	EN	PR

Master Physik

The modules listed here can be found in the [Module Handbook of the Master Physics program](#). Please be aware of the contentual prerequisites of the respective modules given in the module handbook. If a module is listed here, this does *not* automatically mean that the module can be completed with the previous knowledge from the Bachelor of Computer Science alone.

Title	Responsible Persons	Cycle	ECTS	Lang.	Area
Advanced Quantum Information Theory	PD Dr. H. Kampermann	irregular	4 CP	EN	TH
Journal Club on Quantum Information Theory	Prof. Dr. D. Bruß	every semester	3 CP	EN	TH
Quantum Cryptography	PD Dr. H. Kampermann	irregular	4 CP	EN	TH
Surface Code Quantum Computation	PD Dr. H. Kampermann	irregular	6 CP	EN	TH
Theoretical Quantum Optics and Quantum Information	Prof. Dr. D. Bruß	every summer semester	6 CP	EN	TH

Master Betriebswirtschaftslehre

Please note that these modules can *only* be recognized in the "Individual Specialization" area. Please also note the contentual prerequisites for the courses.

Title	Responsible Persons	Cycle	ECTS	Lang.
New Enterprises	Dr. M. Henn	every summer semester	8 CP	DE

Individual Research Project

The project work is usually carried out in the working group in which the Master's thesis will later be written. Every Working group builds this module in such a way that it best prepared for a Master's thesis. Students are also involved in the research of the working group. The project work can also be carried out as group work, provided that the individual performance can be determined adequately.

The following components are typical of project work:

- seminars/project presentations
- individual design and programming tasks
- conducting experiments or evaluations
- reading and processing of specialist literature
- participation in research projects

Module Description

ECTS credits	Total hours	Contact hours	Self-study hours
20 CP	600 hours	nach Vereinbarung	nach Vereinbarung (überwiegender Anteil)
Components	Cycle	Course of Study	Language of instruction
-	je nach Arbeitsgruppe, mindestens jährlich	Master Computer Science	German English

Content

In this module, students should be able to carry out independent scientific work and prepare for the master's thesis. The content therefore depends very much on the respective subject area and the areas of interest of the person.

Learning Outcomes

After completing the course, students are able to

- summarize and illustrate the important concepts of the subject area of the project or future master's thesis
- identify important articles and books for specific problems in the field
- question concepts of the subject and decide which ones are suitable for a future master's thesis
- describe and apply the scientific methods of the department

Bibliography

- In agreement with the supervisor.

Module compatibility

- Projektarbeit

Prerequisites

- None

Conditions for awarding credit points

- The specific requirements depend on the task to be processed. Therefore, the criteria for acquiring the credit points must be determined individually at the beginning of the project work.

Responsible persons

Die Dozierenden der Informatik sowie gegebenenfalls der Mathematisch-Naturwissenschaftlichen Schwerpunktfächer

Master's Thesis

Module Description

ECTS credits	Total hours	Contact hours	Self-Study hours
30 CP	900 hours	in consultation with supervisor	in consultation with supervisor
Format	Cycle	Course of Study	Language of instruction
-	at any time	Master Computer Science	German English

Content

The content of the Master's thesis lies in the selected major

Learning Outcomes

With the written thesis, students should prove that they are able to independently analyse a topic from their chosen major subject within a given period (of 6 months) using scientific methods as well as to concisely develop and competently evaluate or interpret the findings. The Master's thesis must be written in German or English and presented in orally.

Literatur

- in consultation with supervisor

Verwendbarkeit des Moduls

- Master-Arbeit

Prerequisites

To register for the Master's thesis, at least 60 of the credit points to be acquired in the master's program must be proven.

The topic of the Master's thesis is chosen from the field of the chosen Major subject. For this purpose, the modules in the Major should usually be successfully completed.

Conditions for awarding credit points

- Successful processing of the topic and presentation in a draft submitted on time (master's thesis)
- Presentation in an oral presentation with discussion

Modulbeauftragte und hauptamtlich Lehrende

Lecturers in computer science and the mathematical and natural sciences subjects that can be selected as a major

Modules no longer offered

In this chapter you will find all modules that we no longer offer.

Big-Data Systems and Applications

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	every winter semester	Master Computer Science	German

Content

- Infrastructure (Cloud, Edge, Fog)
- Scaling basics (Replication, Consistency, CAP)
- Distributed Hash Tables (Chord and Pastry)
- Konsensus services (Paxos, Zookeeper)
- Key-value storage (Dynamo, Redis, DXRAM)
- Distributed file systems (Google File System)
- NoSQL storage (Google BigTable)
- MapReduce (Basics and examples)
- In-Memory processing (Spark architecture and examples)
- Stream processing (Basics, Spark, Flink)
- Graph processing (Pregel, GraphX)
- Message broker (Pub/Sub, Kafka)

Learning Outcomes

After completing the course, students are able to

- explain concepts and functions of big-data systems
- select and apply big-data frameworks for given problems.
- develop and apply data analytics on data streams with tools used in the exercises

Bibliography

- Relevant literature is provided during the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formally: Bachelor students must meet the requirements for the anticipation of Master modules.

Conditions for awarding credit points

- Successfull pariticipation in the exercices which include three exercise sheets and a project (workload about nine exercise sheets)
- Passing the exam (homework)

Responsible persons

Prof. Dr. Michael Schöttner

Introduction to Linear Optimization

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	jedes Wintersemester	Master Computer Science	English

Content

The lecture covers the basic theory and applications of (integer) linear optimization. In the exercises the content of the lecture is applied and deepened. For that the exercises contain theoretical as well as practical elements. The students use linear and integer linear programming modeling software and solvers to solve applied programming exercises.

- Foundations of Linear Programming
- Linear Programs and their geometric interpretation
- Duality
- The Simplex method
- Integer Linear Programming
- Linear programming-based Branch-and-Bound
- Cutting planes and Branch-and-Cut
- Network flows
- Applications: Selected applications of linear optimization techniques

Learning Outcomes

After completing the course, students are able to

- state the basic principles of linear and integer linear programming (LP and ILP)
- build practical LP and ILP models
- design and implement algorithms that solve these models

Bibliography

- Dimitris Bertsimas, John N. Tsitsiklis: Introduction to Linear Optimization. Athena Scientific. Belmont, MA, USA, 1997.
- Laurence Wolsey: Integer Programming. Wiley. New York, NY, USA, 1998.

Module compatibility

- Elective Area Theoretical Computer Science
- Elective Area Practical or Technical Computer Science
- Elective Area Master study programme Artificial Intelligence and Data Science

Prerequisites

- Formally: Bachelor's students must meet the requirements for the anticipation of Master's modules.

Conditions for awarding credit points

- active participation

- successful completion of exercise sheets
- passing written exam

Responsible persons

Prof. Dr. Gunnar W. Klau

Judgment Aggregation

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	60 hours	90 hours
Components	Cycle	Course of Study	Language of instruction
Lecture (2 HPW) Tutorial (2 HPW)	irregular	Master Computer Science	German

Remarks

- 7.5 CP for students studying according to PO 2005.

Content

This module deals with the aggregation of individual judgments over possibly logically connected propositions. Contents of this lecture are impossibility results and the examination of different methods with their axiomatic and algorithmic properties.

- judgment aggregation procedures (i.e., premise-based, conclusion-based, distance-based)
- axiomatic properties and impossibility results
- agenda characterizations
- different forms of strategic interference (manipulation, bribery, control)

Learning Outcomes

After completing the course, students are able to

- conduct methods for aggregation individual judgments,
- retrace theoretical proofs for impossibility results from the literature,
- name conclusions of the impossibility results in different areas,
- develop new ways of aggregation individual judgments and investigate their properties, and
- give recommendations for specific application areas.

Bibliography

- Jörg Rothe, Dorothea Baumeister, Claudia Lindner und Irene Rothe: Einführung in Computational Social Choice. Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen. Spektrum, Akademischer Verlag. 2011.

Module compatibility

- Elective Area Theoretical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Contentual: Contents of module *Theoretical Computer Science*
- Formal: Bachelor's students must meet the requirements for anticipating Master's modules.

Conditions for awarding credit points

- active and successful participation in the theoretical exercise courses
- written test (exam, usually 90 minutes)

Responsible persons

apl. Prof. Dr. Dorothea Baumeister

Master's Seminar: System Software for Big Data Computing

ECTS credits	Total hours	Contact hours	Self-study hours
5 CP	150 hours	30 hours	120 hours
Components	Cycle	Course of Study	Language of instruction
Seminar (2 HPW)	every summer semester	Master Computer Science	German

Content

- big-data systems: batch-, in-memory, stream-, Map-Reduce-Processing; distributed machine learning, ...
- cloud systems: storage solutions, virtualization, messaging and coordination services,
- operating systems for manycore, NVRAM (persistent memory), RDMA (Remote Direct Memory Access), GPUs, ...

Learning Outcomes

After completing the course, students are able to

- research relevant contents of a given topic and to reproduce them in a written and oral form,
- evaluate the readability and technical quality of other papers,
- plan and hold a presentation,
- ask and answer adequate questions, and
- give and take constructive feedback.

Bibliography

- Relevant literature is provided during the course.

Module compatibility

- Elective Area Practical or Technical Computer Science
- Major Subject
- Individual Supplement
- Application subject for Minor area Master study programme Mathematics

Prerequisites

- Formally: Bachelor's students must meet the requirements for the anticipation of Master's modules.

Conditions for awarding credit points

- Writing a paper for the given topic
- Contributing to the peer review process
- Presentation of the given topic
- Participate in discussions

Responsible persons

Prof. Dr. Michael Schöttner