UNIVERSITÄT DES SAARLANDES

Faculty of Mathematics and Computer Science

Module Descriptions

Data Science and Artificial Intelligence MSc

17th December 2024

List of module categories and modules

Module Category 1

Stammvorlesungen Informatik

Algorithms and Data Structures — AlgoDat

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Kurt Mehlhorn

lecturers Prof. Dr. Raimund Seidel
Prof. Dr. Kurt Mehlhorn

entrance requirements For graduate students: C, C++, Java

assessments / exams

- Regular attendance of classes and tutorials
- Passing the midterm and the final exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

The students know standard algorithms for typical problems in the area's graphs, computational geometry, strings and optimization. Furthermore, they master a number of methods and data-structures to develop efficient algorithms and analyze their running times.

content

- graph algorithms (shortest path, minimum spanning trees, maximal flows, matchings, etc.)
- computational geometry (convex hull, Delaunay triangulation, Voronoi diagram, intersection of line segments, etc.)
- strings (pattern matching, suffix trees, etc.)
- generic methods of optimization (tabu search, simulated annealing, genetic algorithms, linear programming, branch-and-bound, dynamic programming, approximation algorithms, etc.)
- data-structures (Fibonacci heaps, radix heaps, hashing, randomized search trees, segment trees, etc.)
- methods for analyzing algorithms (amortized analysis, average-case analysis, potential methods, etc.

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Compiler Construction CC

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	4	at least every two years	1 semester	6	9

responsible Prof. Dr. Sebastian Hack

lecturers Prof. Dr. Sebastian Hack

entrance requirements For graduate students: none

assessments / exams

- Regular attendance of classes and tutorials
- Written exam at the end of the course, theoretical exercises, and compiler-laboratory project.
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

The students learn, how a source program is lexically, syntactically, and semantically analyzed, and how they are translated into semantically equivalent machine programs. They learn how to increase the efficiency by semantics-preserving transformations. They understand the automata-theoretic foundations of these tasks and learn, how to use the corresponding tools.

content

Lexical, syntactic, semantic analysis of source programs, code generation for abstract and real machines, efficiency-improving program transformations, foundations of program analysis.

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Complexity Theory CT

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	4	at least every two years	1 semester	6	9

responsible Prof. Dr. Markus Bläser

lecturers Prof. Dr. Raimund Seidel
Prof. Dr. Markus Bläser

entrance requirements undergraduate course on theory of computation (e.g. *Grundzüge der Theoretischen Informatik*) is highly recommend.

assessments / exams
- Regular attendance of classes and tutorials
- assignments
- exams (written or oral)

course types / weekly hours
```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload
```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be calculated from the results in the assignments and/or exams, as announced by the Lecturer at the beginning of the course

language English

aims / competences to be developed

The aim of this lecture is to learn important concepts and methods of computational complexity theory. The student shall be enabled to understand recent topics and results in computational complexity theory.

content

Relation among resources like time, space, determinism, nondeterminism, complexity classes, reduction and completeness, circuits and nonuniform complexity classes, logarithmic space and parallel complexity classes, Immerman-Szelepcsenyi theorem, polynomial time hierarchy, relativization, parity and the polynomial methods, Valiant-Vazirani theorem, counting problems and classes, Toda's theorem, probabilistic computations, isolation lemma and parallel algorithms for matching, circuit identity testing, graph isomorphism and interactive proofs.

literature & reading

Arora, Barak: Computational Complexity – A Modern Approach, Cambridge University Press
Oded Goldreich: Computational Complexity – A Conceptual Approach, Cambridge University Press
Dexter Kozen: Theory of Computation, Springer
Schöning, Pruim: Gems of Theoretical Computer Science, Springer

Computer Algebra CA

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Frank-Olaf Schreyer

lecturers Prof. Dr. Frank-Olaf Schreyer

entrance requirements For graduate students: none

assessments / exams
- Regular attendance of classes and tutorials
- Solving the exercises, passing the midterm and the final exam.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

Solving problems occuring in computer algebra praxis
The theory behind algorithms

content

Arithmetic and algebraic systems of equations in geometry, engineering and natural sciences

- integer and modular arithmetics, prime number tests
- polynomal arithmetics and factorization
- fast Fourier-transformation, modular algorithms
- resultants, Gröbnerbasen
- homotopy methods for numerical solving
- real solutions, Sturm chains and other rules for algebraic signs Arithmetic and algebraic systems of equations in geometry, engineering and natural sciences
- integer and modular arithmetics, prime number tests
- polynomal arithmetics and factorization
- fast Fourier-transformation, modular algorithms
- resultants, Gröbnerbasen
- homotopy methods for numerical solving
- real solutions, Sturm chains and other rules for algebraic signs

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Computer Graphics CG

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	4	at least every two years	1 semester	6	9

responsible Prof. Dr. Philipp Slusallek

lecturers Prof. Dr. Philipp Slusallek

entrance requirements Solid knowledge of linear algebra is recommended.

assessments / exams

- Successful completion of weekly exercises (30% of final grade)
- Successful participation in rendering competition (10%)
- Mid-term written exam (20%, final exam prerequisite)
- Final written exam (40%)
- In each of the above a minimum of 50% is required to pass

A re-exam typically takes place during the last two weeks before the start of lectures in the following semester.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade The grade is derived from the above assessments. Possible changes will be announced at the beginnning of each semester.

language English

aims / competences to be developed

This course provides the theoretical and practical foundation for computer graphics. It gives a wide overview of topics, techniques, and approaches used in various aspects of computer graphics but has some focus on image synthesis or rendering. The first part of the course uses ray tracing as a driving applications to discuss core topics of computer graphics, from vector algebra all the way to sampling theory, the human visual system, sampling theory, and spline curves and surfaces. A second part then uses rasterization approach as a driving example, introducing the camera transformation, clipping, the OpenGL API and shading langue, plus advanced techniques.

As part of the practical exercises the students incrementally build their own ray tracing system. Once the basics have been covered, the students participate in a rendering competition. Here they can implement their favorite advanced algorithm and are asked to generate a high-quality rendered image that shows their techniques in action.

content

- Introduction
- Overview of Ray Tracing and Intersection Methods
- Spatial Index Structures
- Vector Algebra, Homogeneous Coordinates, and Transformations
- Light Transport Theory, Rendering Equation
- BRDF, Materials Models, and Shading
- Texturing Methods
- Spectral Analysis, Sampling Theory
- Filtering and Anti-Aliasing Methods

- Recursive Ray Tracing & Distribution Ray-Tracing
- Human Visual System & Color Models
- Spline Curves and Surfaces
- Camera Transformations & Clipping
- Rasterization Pipeline
- OpenGL API & GLSL Shading
- Volume Rendering (opt.)

literature & reading

Will be announced in the lecture.

Continuous Optimization — OPT

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Peter Ochs

lecturers Prof. Dr. Peter Ochs

entrance requirements Undergraduate mathematics (e.g. *Mathematik für Informatiker I*, *II* and *III*) and some elementary programming knowledge is recommended.

assessments / exams

- Regular attendance of classes and tutorials
- Solving accompanying exercises
- Successful partcipation in the final or re-exam

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

After taking this course, students will have an overview of classical optimization methods and analysis tools for continuous optimization problems, which allows them to model and solve practical problems. Moreover, in the tutorials, some experience will be gained to implement and numerically solve practical problems.

content

1. Introduction
   - Mathematical Optimization
   - Applications
   - Performance of Numerical Methods
   - Existence of a Solution
   - The Class of Convex Optimization Problems
2. Unconstrained Optimization
   - Optimality Conditions
   - Descent Methods
   - Gradient Descent Method
   - Conjugate Gradient Method
   - Newton's Method
   - Quasi-Newton Methods
   - Gauss-Newton Method
   - Computing Derivatives
3. Constrained Optimization
   - Motivation

- Optimality Conditions for Constrained Problems
- Method of Feasible Directions
- Linear Programming
- Quadratic Programming
- Sequential Quadratic Programming (SQP)
- Penalty and Barrier Methods

literature & reading

- J. Nocedal und S. J. Wright: Numerical Optimization. Springer, 2006.
- F. Jarre und J. Stoerr: Optimierung. Springer, 2004.
- D. Bertsekas: Nonlinear Programming. Athena Scientific, 1999.
- Y. Nesterov: Introductory Lectures on Convex Optimization - A Basic Course. Kluwer Academic Publisher, 2004.
- T. Rockafellar and R. J.-B. Wets: Variational Analysis. Springer-Verlag Berlin Heidelberg, 1998.

Convex Analysis and Optimization CAO

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Peter Ochs

lecturers Prof. Dr. Peter Ochs

entrance requirements Undergraduate mathematics (e.g. *Mathematik für Informatiker I*, *II* and *III*) and some elementary programming knowledge is recommended.

assessments / exams

- Regular attendance of classes and tutorials
- Solving accompanying exercises
- Successful participation in the final or re-exam

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

After taking the course, students know about the most relevant concepts of convex analysis and convex optimization. They are able to read and understand related scientific literature. Moreover, they can rate the difficulty of convex optimization problems arising in applications in machine learning or computer vision and select an efficient algorithm accordingly. Moreover, they develop basic skills in solving practical problems with Python.

content

1. Introduction
   - Introduction
   - Applications
2. Convex Geometry
   - Foundations
   - Convex Feasibility Problems
3. Convex Analysis Background
   - Preliminaries
   - Convex Functions
4. Smooth Convex Optimization
   - Optimality Conditions
   - Gradient Descent Method
   - Lower complexity bounds
   - Accelerated and Inertial Algorithms

5. Non-smooth Convex Analysis
    - Continuity of Convex Functions
    - Convexity from Epigraphical Operations
    - The Subdifferential
6. Non-smooth Convex Optimization
    - Fermat's Rule
    - Duality in Optimization and Primal / Dual Problems
    - Algorithms
    - Lower complexity bounds
    - Saddle Point Problems

literature & reading

- T. Rockafellar: Convex Analysis. Princeton University Press, 1970.
- Y. Nesterov: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic Publishers, 2004.
- D.P. Bertsekas: Convex Analysis and Optimization. Athena Scientific, 2003.
- S. Boyd: Convex Optimization. Cambridge Univeristy Press, 2004.
- H. H. Bauschke and P. L. Combettes: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, 2011.
- T. Rockafellar and R. J.-B. Wets: Variational Analysis. Springer-Verlag Berlin Heidelberg, 1998.

Cryptography Crypto

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Dr. Nico Döttling

lecturers Prof. Dr. Cas Cremers
Dr. Nico Döttling
Dr. Antoine Joux
Dr. Lucjan Hanzlik
Dr. Julian Loss

entrance requirements For graduate students: Basic knowledge in theoretical computer science required, background knowledge in number theory and complexity theory helpful

assessments / exams
- Oral / written exam (depending on the number of students)
- A re-exam is normally provided (as written or oral examination).

course types / weekly hours
```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload
```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

The students will acquire a comprehensive knowledge of the basic concepts of cryptography and formal definitions. They will be able to prove the security of basic techniques.

content

- Symmetric and asymmetric encryption
- Digital signatures and message authentication codes
- Information theoretic and complexity theoretic definitions of security, cryptographic reduction proofs
- Cryptographic models, e.g. random oracle model
- Cryptographic primitives, e.g. trapdoor-one-way functions, pseudo random generators, etc.
- Cryptography in practice (standards, products)
- Selected topics from current research

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Cyber-Physical Systems

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Martina Maggio

lecturers Prof. Dr. Martina Maggio

entrance requirements none

assessments / exams

- Written exam at the end of the course.
- A re-exam takes place before the start of the following semester.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorials
= 6 h (weekly)
```

total workload

```
   75 h lectures
+  15 h mandatory assignments
+ 180 h individual study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams and assignments. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

By completing the Cyber-Physical Systems course, students will acquire the ability to model, analyze, control, and implement embedded systems that interact with the physical world, equipping them to design reliable and efficient systems for a variety of applications in modern technology.

content

Cyber-Physical Systems are embedded systems that integrate computation with physical processes. These systems are ubiquitous in our daily lives, powering technologies such as smart watches, household appliances, mobile phones, and automotive control systems. In fact, the majority of modern computing devices are embedded systems, with an estimated 98% of new CPUs being embedded in larger systems.

This course provides a comprehensive foundation for understanding, designing, and programming cyber-physical systems, emphasizing their theoretical and practical aspects. It is structured into three interconnected parts:

1. *Models:* Students will learn how to represent the physical systems that embedded systems interact with, exploring dynamical systems in both continuous and discrete time. Additionally, the course will briefly introduce more advanced models, which combine discrete state systems with dynamical systems.
2. *Control:* This module focuses on principles for modifying the behavior of physical systems through computation. Students will study and apply control techniques such as state feedback and PID control, learning how these methods influence the interaction between embedded systems and their environments.
3. *Implementation:* The final course part addresses practical challenges in embedded systems programming. Topics include scheduling, communication, and fault tolerance. This ensures that students are equipped to implement robust and efficient embedded systems in real-world scenarios.

By the end of this course, students will possess the skills needed to design and implement cyber-physical systems that meet specific functional and performance requirements, preparing them for roles in cutting-edge industries where embedded systems play a critical role, such as the automotive industry and for research in the cyber-physical systems domain.

literature & reading

Will be announced before the start of the course on the course page on the Internet.

additional information

This module was formerly also known as *Embedded Systems*.

Data Networks

DN

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr.-Ing. Holger Hermanns

lecturers Prof. Dr.-Ing. Holger Hermanns
Prof. Dr. Anja Feldmann

entrance requirements For graduate students: none

assessments / exams

- Regular attendance of classes and tutorials
- Qualification for final exam through mini quizzes during classes
- Possibility to get bonus points through excellent homework
- Final exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

After taking the course students have

- a thorough knowledge regarding the basic principles of communication networks,
- the fundamentals of protocols and concepts of protocol,
- Insights into fundamental motivations of different pragmatics of current network solutions,
- Introduction to practical aspects of data networks focusing on internet protocol hierarchies

content

Introduction and overview

Cross section:

- Stochastic Processes, Markov models,
- Fundamentals of data network performance assessment
- Principles of reliable data transfer
- Protokols and their elementary parts
- Graphs and Graphalgorithms (maximal flow, spanning tree)
- Application layer:
- Services and protocols
- FTP, Telnet
- Electronic Mail (Basics and Principles, SMTP, POP3, ..)
- World Wide Web (History, HTTP, HTML)

- Transport Layer:
- Services and protocols
- Addressing
- Connections and ports
- Flow control
- QoS
- Transport Protocols (UDP, TCP, SCTP, Ports)
- Network layer:
- Services and protocols
- Routing algorithms
- Congestion Control
- Addressing
- Internet protocol (IP)
- Data link layer:
- Services and protocols
- Medium access protocols: Aloha, CSMA (-CD/CA), Token passing
- Error correcting codes
- Flow control
- Applications: LAN, Ethernet, Token Architectures, WLAN, ATM
- Physical layer
- Peer-to-Peer and Ad-hoc Networking Principles

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Digital Transmission & Signal Processing DTSP

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr.-Ing. Thorsten Herfet

lecturers Prof. Dr.-Ing. Thorsten Herfet

entrance requirements The lecture requires a solid foundation of mathematics (differential and integral calculus) and probability theory. The course will, however, refresh those areas indispensably necessary for telecommunications and potential intensification courses and by this open this potential field of intensification to everyone of you.

assessments / exams Regular attendance of classes and tutorials
Passing the final exam in the 2nd week after the end of courses.
Eligibility: Weekly exercises / task sheets, grouped into two blocks corresponding to first and second half of the lecture. Students must provide min. 50% grade in each of the two blocks to be eligible for the exam.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Final exam mark

language English

aims / competences to be developed

Digital Signal Transmission and Signal Processing refreshes the foundation laid in "Signals and Systems" [Modulkennung]. Including, however, the respective basics so that the various facets of the introductory study period (Bachelor in Computer Science, Vordiplom Computer- und Kommunikationstechnik, Elektrotechnik or Mechatronik) and the potential main study period (Master in Computer Science, Diplom-Ingenieur Computer- und Kommunikationstechnik or Mechatronik) will be paid respect to.

content

As the basic principle, the course will give an introduction into the various building blocks that modern telecommunication systems do incorporate. Sources, sinks, source and channel coding, modulation and multiplexing are the major keywords, but we will also deal with dedicated pieces like A/D- and D/A-converters and quantizers in a little bit more depth.

The course will refresh the basic transformations (Fourier, Laplace) that give access to system analysis in the frequency domain, it will introduce derived transformations (z, Hilbert) for the analysis of discrete systems and modulation schemes and it will briefly introduce algebra on finite fields to systematically deal with error correction schemes that play an important role in modern communication systems.

literature & reading

Will be announced before the start of the course on the course page on the Internet.

additional information

This module was formerly also known as *Telecommunications I*.

Distributed Systems — DS

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	4	at least every two years	1 semester	6	9

responsible Prof. Peter Druschel, Ph.D.

lecturers Prof. Peter Druschel, Ph.D.
Allen Clement, Ph.D

entrance requirements *Operating Systems* or *Concurrent Programming*

assessments / exams

- Regular attendance at classes and tutorials.
- Successful completion of a course project in teams of 2 students. (Project assignments due approximately every 2 weeks.)
- Passing grade on 2 out of 3 written exams: midterm, final exam, and a re-exam that takes place during the last two weeks before the start of lectures in the following semester.
- Final course grade: 50% project, 50% best 2 out of 3 exams.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

Introduction to the principles, design, and implementation of distributed systems.

content

- Communication: Remote procedure call, distributed objects, event notification, Inhalt dissemination, group communication, epidemic protocols.
- Distributed storage systems: Caching, logging, recovery, leases.
- Naming. Scalable name resolution.
- Synchronization: Clock synchronization, logical clocks, vector clocks, distributed snapshots.
- Fault tolerance: Replication protocols, consistency models, consistency versus availability trade-offs, state machine replication, consensus, Paxos, PBFT.
- Peer-to-peer systems: consistent hashing, self-organization, incentives, distributed hash tables, Inhalt distribution networks.
- Data centers. Architecture and infrastructure, distributed programming, energy efficiency.

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Geometric Modelling GM

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	4	at least every two years	1 semester	6	9

responsible Prof. Dr. Hans-Peter Seidel

lecturers Prof. Dr. Hans-Peter Seidel
Dr. Rhaleb Zayer

entrance requirements calculus and basic programming skills

assessments / exams

- Regular attendance and participation.
- Weekly Assignments (10% bonus towards the course grade; bonus points can only improve the grade; they do not affect passing)
- Passing the written exams (mid-term and final exam).
- The mid-term and the final exam count for 50% each, but 10% bonus from assignments will be added.
- A re-exam takes place at the end of the semester break or early in the next semester.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

Practical assignments in groups of 3 students (practice)
Tutorials consists of a mix of theoretical + practical assignments.

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be based on the performance in exams, exercises and practical tasks. The detailed terms will be announced by the module coordinator.

language English

aims / competences to be developed

Gaining knowledge of the theoretical aspect of geometric modelling problems, and the practical solutions used for modelling and manipulating curves and surfaces on a computer. From a broader perspective: Learning how to represent and interact with geometric models in a discretized, digital form (geometric representations by functions and samples; design of linear function spaces; finding “good” functions with respect to a geometric modelling task in such spaces).

content

- Differential geometry Fundamentals
- Interpolation and Approximation
- Polynomial Curves
- Bezier and Rational Bezier Curves
- B-splines, NURBS
- Spline Surfaces
- Subdivision and Multiresolution Modelling
- Mesh processing
- Approximation of differential operators
- Shape Analysis and Geometry Processing

literature & reading

Will be announced before the term begins on the lecture website.

Human Computer Interaction HCI

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Jürgen Steimle

lecturers Prof. Dr. Jürgen Steimle

entrance requirements undergraduate students: *Programmierung 1* and *2*
graduate students: none

assessments / exams Regular attendance of classes and tutorials
Successful completion of exercises and course project
Final exam
A re-exam takes place (as written or oral examination).

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

This course teaches the theoretical and practical foundations for human computer interaction. It covers a wide overview of topics, techniques and approaches used for the design and evaluation of modern user interfaces.

The course covers the principles that underlie successful user interfaces, provides an overview of input and output devices and user interface types, and familiarizes students with the methods for designing and evaluating user interfaces. Students learn to critically assess user interfaces, to design user interfaces themselves, and to evaluate them in empirical studies.

content

- Fundamentals of human-computer interaction
- User interface paradigms, input and output devices
- Desktop & graphical user interfaces
- Mobile user interfaces
- Natural user interfaces
- User-centered interaction design
- Design principles and guidelines
- Prototyping

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Introduction to Computational Logic — ICL

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	4	at least every two years	1 semester	6	9

responsible Prof. Dr. Gert Smolka

lecturers Prof. Dr. Gert Smolka

entrance requirements none

assessments / exams
- Regular attendance of classes and tutorials.
- Passing the midterm and the final exam.

course types / weekly hours
```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload
```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

- structure of logic languages based on type theory
- distinction notation / syntax / semantics
- structure and formal representation of mathematical statements
- structure and formal representation of proofs (equational and natural deduction)
- solving Boolean equations
- proving formulas with quantifiers
- implementing syntax and deduction

content

Type Theory:

- functional representation of mathematical statements
- simply typed lambda calculus, De Bruijn representation and substitution, normalization, elimination of lambdas
- Interpretations and semantic consequence
- Equational deduction, soundness and completeness
- Propositional Logic
- Boolean Axioms, completeness for 2-valued interpretation
- resolution of Boolean equations, canonical forms based on decision trees and resolution

Predicate Logic (higher-order):

- quantifier axioms
- natural deduction
- prenex and Skolem forms

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Operating Systems

OS

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible	Prof. Peter Druschel, Ph.D.
lecturers	Prof. Peter Druschel, Ph.D. Björn Brandenburg, Ph.D
entrance requirements	For graduate students: none
assessments / exams	Regular attendance at classes and tutorials Successful completion of a course project in teams of 2 students Passing 2 written exams (midterm and final exam) A re-exam takes place during the last two weeks before the start of lectures in the following semester.
course types / weekly hours	4 h lectures + 2 h tutorial = 6 h (weekly)
total workload	90 h of classes + 180 h private study = 270 h (= 9 ECTS)
grade	Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.
language	English

aims / competences to be developed

Introduction to the principles, design, and implementation of operating systems

content

Process management:

- Threads and processes, synchronization
- Multiprogramming, CPU Scheduling
- Deadlock

Memory management:

- Dynamic storage allocation
- Sharing main memory
- Virtual memory

I/O management:

- File storage management
- Naming
- Concurrency, Robustness, Performance

Virtual machines

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Optimization

Opti

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Kurt Mehlhorn

lecturers Prof. Dr. Kurt Mehlhorn
Dr. Andreas Karrenbauer

entrance requirements For graduate students: none

assessments / exams

- Regular attendance of classes and tutorials
- Solving accompanying exercises, successful partcipation in midterm and final exam
- Grades: Yes
- The grade is calculated from the above parameters according to the following scheme: 20%, 30%, 50%
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

The students learn to model and solve optimization problems from theory as from the real world

content

Linear Programming: Theory of polyhedra, simplex algorithm, duality, ellipsoid method * Integer linear programming: Branch-and-Bound, cutting planes, TDI-Systems * Network flow: Minimum cost network flow, minimum mean cycle cancellation algorithm, network simplex method * Matchings in graphs: Polynomial matching algorithms in general graphs, integrality of the matching polytope, cutting planes * Approximation algorithms: LP-Rounding, greedy methods, knapsack, bin packing, steiner trees and forests, survivable network design

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Semantics

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible	Prof. Dr. Gert Smolka
lecturers	Prof. Dr. Gert Smolka
entrance requirements	For graduate students: core lecture Introduction to Computational Logic
assessments / exams	• Regular attendance of classes and tutorials. • Passing the midterm and the final exam

course types / weekly hours	`  4 h lectures`
	`+ 2 h tutorial`
	`= 6 h (weekly)`

total workload	`   90 h of classes`
	`+ 180 h private study`
	`= 270 h (= 9 ECTS)`

grade	Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.
language	English

aims / competences to be developed

Understanding of

- Logical structure of programming languages
- Formal models of programming languages
- Type and module systems for programming languages

content

Theory of programming languages, in particular:

- Formal models of functional and object-oriented languages
- Lambda Calculi (untyped, simply typed, System F, F-omega, Lambda Cube, subtyping, recursive types, Curry-Howard Correspondence)
- Algorithms for type checking and type reconstruction

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Software Engineering — SE

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Sven Apel

lecturers Prof. Dr. Sven Apel

entrance requirements

- Knowledge of programming concepts (as taught in the lectures *Programmierung 1* and *Programmierung 2*)
- Basic knowledge of software processes, design, and testing (as taught and applied in the lecture *Softwarepraktikum*)

assessments / exams Beside the lecture and weekly practical exercises, there will be a number of assignments in the form of mini-projects for each student to work on (every two to three weeks). The assignments will be assessed based on the principles covered in the lecture. Passing all assignments is a prerequisite for taking the final written exam. The final grade is determined only by the written exam. Further examination details will be announced by the lecturer at the beginning of the course. In short:

- Passing all assignments (prerequisite for the written exam)
- Passing the written exam

course types / weekly hours

```
  4 h lectures
+ 2 h exercises
= 6 h (weekly)
```

total workload

```
   90 h of classes and exercises
+ 180 h private study and assignments
= 270 h (= 9 ECTS)
```

grade The grade is determined by the written exam. Passing all assignments is a prerequisite for taking the written exam. The assignments do not contribute to the final grade. Further examination details will be announced by the lecturer at the beginning of the course.

language English

aims / competences to be developed

- The students know and apply modern software development techniques.
- They are aware of key factors contributing to the complexity of real-world software systems, in particular, software variability, configurability, feature interaction, crosscutting concerns, and how to address them.
- They know how to apply established design and implementation techniques to master software complexity.
- They are aware of advanced design and implementation techniques, including collaboration-based design, mixins/traits, aspects, pointcuts, advice.
- They are aware of advanced quality assurance techniques that take the complexity of real-world software systems into account: variability-aware analysis, sampling, feature-interaction detection, predictive performance modeling, etc.
- They appreciate the role of non-functional properties and know how to predict and optimize software systems regarding these properties.
- They are able to use formal methods to reason about key techniques and properties covered in the lecture.

content

- Domain analysis, feature modeling
- Automated reasoning about software configuration using SAT solvers
- Runtime parameters, design patterns, frameworks
- Version control, build systems, preprocessors
- Collaboration-based design
- Aspects, pointcuts, advice
- Expression problem, preplanning problem, code scattering & tangling, tyranny of the dominant decomposition, inheritance vs. delegation vs. mixin composition
- Feature interaction problem (structural, control- & data-flow, behavioral, non-functional feature interactions)
- Variability-aware analysis and variational program representation (with applications to type checking and static program analysis)
- Sampling (random, coverage)
- Machine learning for software performance prediction and optimization

literature & reading

- Feature-Oriented Software Product Lines: Concepts and Implementation. S. Apel, et al., Springer, 2013.
- Generative Programming: Methods, Tools, and Applications: Methods, Techniques and Applications. K. Czarnecki, et al., Addison-Wesley, 2000.
- Mastering Software Variability with FeatureIDE. J. Meinicke, et al., Springer, 2017.

Verification

Veri

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr.-Ing. Holger Hermanns

lecturers Prof. Dr.-Ing. Holger Hermanns
Prof. Bernd Finkbeiner, Ph.D

entrance requirements For graduate students: none

assessments / exams

- Regular attendance of classes and tutorials
- Passing the final exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

The students become familiar with the standard methods in computer-aided verification. They understand the theoretical foundations and are able to assess the advantages and disadvantages of different methods for a specific verification project. The students gain first experience with manual correctness proofs and with the use of verification tools.

content

- models of computation and specification languages: temporal logics, automata over infinite objects, process algebra
- deductive verification: proof systems (e.g., Floyd, Hoare, Manna/Pnueli), relative completeness, compositionality
- model checking: complexity of model checking algorithms, symbolic model checking, abstraction case studies

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Module Category 2

Stammvorlesungen DSAI

Artificial Intelligence — AI

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Jörg Hoffmann

lecturers Prof. Dr. Jörg Hoffmann

entrance requirements *Programming 1*, *Programming 2*, *Fundamentals of Data Structures and Algorithms*, and *Elements of Machine Learning* or other courses in machine learning are recommended.

assessments / exams

- Regular attendance of classes and tutorials
- Solving of weekly assignments
- Passing the final written exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from the performance in exams. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

Knowledge about basic methods in Artificial Intelligence

content

Search:

- Uninformed- and informed search procedures
- Monte-Carlo tree search

Planning:

- Formalism and complexity
- Critical-path heuristics
- Delete relaxation heuristics
- Abstraction heuristics

Markov decision processes:

- Discounted reward and expected cost
- Value iteration
- Informed search
- Reinforcement learning

Games:

- Adversarial search
- Learning from self-play

literature & reading

Russel & Norvig Artificial Intelligence: A Modern Approach;
further reading will be announced before the start of the course on the course page on the Internet.

Automated Reasoning

AR

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Christoph Weidenbach

lecturers Prof. Dr. Christoph Weidenbach

entrance requirements *Introduction to Computational Logic*

assessments / exams

- Regular attendance of classes and tutorials
- Weekly assignments
- Practical work with systems
- Passing the final and mid-term exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from performance in exams, exercises and practical tasks. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

The goal of this course is to provide familiarity with logics, calculi, implementation techniques, and systems providing automated reasoning.

content

Propositional Logic – CDCL, Superposition - Watched Literals
First-Order Logic without Equality – (Ordered) Resolution,
Equations with Variables – Completion, Termination
First-Order Logic with Equality – Superposition (SUP) - Indexing

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Database Systems DBS

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Jens Dittrich

lecturers Prof. Dr. Jens Dittrich

entrance requirements especially Saarland University CS department's undergraduate lecture *Big Data Engineering* (former *Informationssysteme*), *Programmierung 1* and *2*, *Algorithmen und Datenstrukturen* as well as *Nebenläufige Programmierung*

For graduate students:

- motivation for databases and database management systems;
- the relational data model;
- relational query languages, particularly relational algebra and SQL;
- **solid** programming skills in Java and/or C++
- undergrad courses in algorithms and data structures, concurrent programming

assessments / exams

- Passing a two-hour written exam at the end of the semester
- Successful demonstration of programming project (teams of up to three students are allowed); the project may be integrated to be part of the weekly assignments

Grades are based on written exam; 50% in weekly assignments (in paper and additionally paper or electronic quizzes) must be passed to participate in the final and repetition exams.

A repetition exam takes place during the last two weeks before the start of lectures in the following semester.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

This class may be run as a flipped classroom, i.e. 2 hours of lectures may be replaced by self-study of videos/papers; the other 2 hours may be used to run a group exercice supervised by the professor called "the LAB")

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined based on project, midterm and best of endterm and reexam.

language English

aims / competences to be developed

Database systems are the backbone of most modern information systems and a core technology without which today's economy – as well as many other aspects of our lifes – would be impossible in their present forms. The course teaches the architectural and algorithmic foundations of modern database management systems (DBMS), focussing on database systems internals rather than applications. Emphasis is made on robust and time-tested techniques that have led databases to be considered a mature technology and one of the greatest success stories in computer science. At the same time, opportunities for exciting research in this field will be pointed out.

In the exercise part of the course, important components of a DBMS will be treated and where possible implemented and their performance evaluated. The goal this is to work with the techniques introduced in the lecture and to understand them and their practical implications to a depth that would not be attainable by purely theoretical study.

content

The course ”Database Systems” will introduce students to the internal workings of a DBMS, in particular:

- storage media (disk, flash, main memory, caches, and any other future storage medium)
- data managing architectures (DBMS, streams, file systems, clouds, appliances)
- storage management (DB-file systems, raw devices, write-strategies, differential files, buffer management)
- data layouts (horizontal and vertical partitioning, columns, hybrid mappings, compression, defragmentation)
- indexing (one- and multidimensional, tree-structured, hash-, partition-based, bulk-loading and external sorting, differential indexing, read- and write-optimized indexing, data warehouse indexing, main-memory indexes, sparse and dense, direct and indirect, clustered and unclustered, main memory versus disk and/or flash-based)
- processing models (operator model, pipeline models, push and pull, block-based iteration, vectorization, query compilation)
- processing implementations (join algorithms for relational data, grouping and early aggregation, filtering)
- query processing (scanning, plan computation, SIMD)
- query optimization (query rewrite, cost models, cost-based optimization, join order, join graph, plan enumeration)
- data recovery (single versus multiple instance, logging, ARIES)
- parallelization of data and queries (horizontal and vertical partitioning, shared-nothing, replication, distributed query processing, NoSQL, MapReduce, Hadoop and/or similar and/or future systems)
- read-optimized system concepts (search engines, data warehouses, OLAP)
- write-optimized system concepts (OLTP, streaming data)
- management of geographical data (GIS, google maps and similar tools)
- main-memory techniques

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Image Processing and Computer Vision IPCV

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Joachim Weickert

lecturers Prof. Dr. Joachim Weickert

entrance requirements Undergraduate mathematics (e.g. Mathematik für Informatiker I-III) and elementary programming knowledge in C

assessments / exams

- For the homework assignments one can obtain up to 24 points per week. Actively participating in the classroom assignments gives 12 more points per week, regardless of the correctness of the solutions. To qualify for both exams one needs 2/3 of all possible points.
- Passing the final exam or the re-exam.
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined from the performance in the exam or the re-exam. The better grade counts.

language English

aims / competences to be developed

Broad introduction to mathematical methods in image processing and computer vision. The lecture qualifies students for a bachelor thesis in this field. Together with the completion of advanced or specialised lectures (9 credits at least) it is the basis for a master thesis in this field.

content

```
Inhalt

1. Basics
   1.1 Image Types and Discretisation
   1.2 Degradations in Digital Images
2. Colour Perception and Colour Spaces
3. Image Transformations
   3.1 Continuous Fourier Transform
   3.2 Discrete Fourier Transform
   3.3 Image Pyramids
   3.4 Wavelet Transform
4. Image Compression
5. Image Interpolation
6. Image Enhancement
   6.1 Point Operations
```

```
   6.2 Linear Filtering and Feature Detection
   6.3 Morphology and Median Filters
   6.3 Wavelet Shrinkage, Bilateral Filters, NL Means
   6.5 Diffusion Filtering
   6.6 Variational Methods
   6.7 Deconvolution Methods
7. Texture Analysis
8. Segmentation
   8.1 Classical Methods
   8.2 Variational Methods
9. Image Sequence Analysis
   9.1 Local Methods
   9.2 Variational Methods
10. 3-D Reconstruction
   10.1 Camera Geometry
   10.2 Stereo
   10.3 Shape-from-Shading
11. Object Recognition
   11.1 Hough Transform
   11.2 Invariants
   11.3 Eigenspace Methods
```

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Information Retrieval and Data Mining — IRDM

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Gerhard Weikum

lecturers Prof. Dr. Gerhard Weikum

entrance requirements Good knowledge of undergraduate mathematics (linear algebra, probability theory) and basic algorithms.

assessments / exams

- Regular attendance of classes and tutor groups
- Presentation of solutions in tutor groups
- Passing 2 of 3 written tests (after each third of the semester)
- Passing the final exam (at the end of the semester)

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined by the performance in written tests, tutor groups, and the final exam. Details will be announced on the course web site.

language English

aims / competences to be developed

The lecture teaches models and algorithms that form the basis for search engines and for data mining and data analysis tools.

content

Information Retrieval (IR) and Data Mining (DM) are methodologies for organizing, searching and analyzing digital Inhalts from the web, social media and enterprises as well as multivariate datasets in these contexts. IR models and algorithms include text indexing, query processing, search result ranking, and information extraction for semantic search. DM models and algorithms include pattern mining, rule mining, classification and recommendation. Both fields build on mathematical foundations from the areas of linear algebra, graph theory, and probability and statistics.

literature & reading

Will be announced on the course web site.

Machine Learning ML

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Isabel Valera

lecturers Prof. Dr. Isabel Valera

entrance requirements The lecture gives a broad introduction into machine learning methods. After the lecture the students should be able to solve and analyze learning problems.

assessments / exams
- Regular attendance of classes and tutorials.
- 50% of all points of the exercises have to be obtained in order to qualify for the exam.
- Passing 1 out of 2 exams (final, re-exam).

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Determined from the results of the exams, exercises and potential projects. The exact grading modalities are announced at the beginning of the course.

language English

aims / competences to be developed

The lecture gives a broad introduction into machine learning methods. After the lecture the students should be able to solve and analyze learning problems.

content

- Bayesian decision theory
- Linear classification and regression
- Kernel methods
- Bayesian learning
- Semi-supervised learning
- Unsupervised learning
- Model selection and evaluation of learning methods
- Statistical learning theory
- Other current research topics

literature & reading

Will be announced before the start of the course on the course page on the Internet.

Neural Networks: Theory and Implementation NNTI

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**every winter semester**	**1 semester**	**6**	**9**

responsible Prof. Dr. Dietrich Klakow

lecturers Prof. Dr. Dietrich Klakow

entrance requirements *Mathemathik für Informatiker I - III* or comparable; good programming skills.

assessments / exams Written Exam

course types / weekly hours
```
  2 h lectures
+ 2 h tutorial
+ 2 h project work
= 6 h (weekly)
```

total workload
```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Written exam and graded projects. Exact details will be announced in the first lecture.

language English

aims / competences to be developed

The participants will be introduced to the key ideas of basic classification algorithms and in particular neural networks. A focus is also the implementation and applications to relevant problems. To achieve this, there will be theoretical exercises as well as project work.

content

- Classification
- Regression
- Linear Classifiers
- Perceptron
- Support Vector Machines
- Multy-Layer Perceptrons
- Deep Learning Software
- Autoencoders
- LSTMs
- Recurrent Neural Networks
- Sequence-to-sequence learning

literature & reading

Ian Goodfellow and Yoshua Bengio and Aaron Courville
Deep Learning
MIT Press, 2016
http://www.deeplearningbook.org

Module Category 3

Vertiefungsvorlesungen DSAI

Ethics for Nerds

E4N

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**occasional / summer semester**	**1 semester**	**4**	**6**

responsible Prof. Dr.-Ing. Holger Hermanns

lecturers Prof. Dr.-Ing. Holger Hermanns
Kevin Baum
Sarah Sterz

entrance requirements We expect basic knowledge of propositional and first-order logic, an open mind, and interest to look at computer science in ways you probably are not used to.

assessments / exams The details of exam admission and grading are announced at the beginning of each iteration. Typically, participant are graded based on

- an exam or a re-exam (the better mark counts),
- a short essay where the participant has to argue for or against a moral claim in a topic from computer science.

To get the exam admission, participants usually have to get 50% of the points on weekly exercise sheets.

course types / weekly hours

```
  2 h lectures
+ 2 h tutorial
= 4 h (weekly)
```

(may be adjusted before the start of each iteration of the course)

total workload

```
   60 h of classes
+ 120 h private study
= 180 h (= 6 ECTS)
```

grade Will be determined based on exam performance, essay performance, and possibly exercise outcomes. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

Many computer scientists will be confronted with morally difficult situations at some point in their career – be it in research, in business, or in industry. This module equips participants with the crucial assets enabling them to recognize such situations and to devise ways to arrive at a justified moral judgment regarding the question what one is permitted to do and what one should better not do. For that, participants will be made familiar with moral theories from philosophy, as well as different Codes of Ethics for computer scientists. Since one can quickly get lost when talking about ethics and morals, it is especially important to talk and argue clearly and precisely. In order to do prepare for that, the module offers substantial training regarding formal and informal argumentation skills enabling participants to argue beyond the level of everyday discussions at bars and parties. In the end, succesful participants are able to assess a morally controversial topic from computer science on their own and give a convincing argument for their respective assessments.

The module is intended to always be as clear, precise, and analytic as possible. What you won't find here is the meaningless bla-bla, needlessly poetic language, and vague and wordy profundity that some people tend to associate with philosophy.

content

This course covers:

- an introduction to the methods of philosophy, argumentation theory, and the basics of normative as well as applied ethics;
- relevant moral codices issued by professional associations like the ACM, the IEEE, and more;
- starting points to evaluate practices and technologies already in use or not that far away, including for instance: filter bubbles and echo chambers, ML-algorithms as predictive tools, GPS-tracking, CCTV and other tools from surveillance, fitness trackers, big data analysis, autonomous vehicles, lethal autonomous weapons systems and so on;
- an outlook on more futuristic topics like machine ethics, roboethics, and superintelligences;
- and more.

The content of the course is updated regularly to always be up-to-date and cover the currently most relevant topics, technologies, policies, and developments.

literature & reading

Will be announced before the start of the course on the course page.

Mathematische Satistik

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**unregelmässig**	**1 Semester**	**6**	**9**

responsible Prof. Dr. Henryk Zähle

lecturers Dozent/inn/en der Mathematik

entrance requirements none

assessments / exams Regelmäßige, aktive Teilnahme an der Vorlesung und an den begleitenden Übungen; Abschlussprüfung

course types / weekly hours

```
  4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS
```

total workload

```
   60 h Vorlesung
+  30 h Übungen
+ 180 h Eigenstudium
        (Vor- und Nachbereitung, Bearbeitung Übungsaufgaben)
= 270 h (= 9 ECTS)
```

grade Durch Klausur(en) und mündliche Prüfung. Der Modus wird zu Beginn der Vorlesung bekannt gegeben.

language Deutsch

aims / competences to be developed

content

- Statistische Modelle; Modellwahl und Modellüberprüfung
- Punktschätzungen
- Bereichsschätzungen
- Hypothesentests

literature & reading

Bekanntgabe jeweils vor der Vorlesung auf der Vorlesungsseite im Internet

additional information

Methoden: Information durch Vorlesung; Vertiefung durch Eigentätigkeit (Nacharbeit, Übungen).

Anmeldung: Bekanntgabe jeweils rechtzeitig vor Semesterbeginn durch Aushang und im Internet.

Security | Sec

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

responsible Prof. Dr. Michael Backes

lecturers Prof. Dr. Michael Backes
Prof. Dr. Cas Cremers

entrance requirements For graduate students: none

assessments / exams

- Regular attendance of classes and tutorials
- Passing the final exam
- A re-exam is normally provided (as written or oral examination).

course types / weekly hours

```
  4 h lectures
+ 2 h tutorial
= 6 h (weekly)
```

total workload

```
   90 h of classes
+ 180 h private study
= 270 h (= 9 ECTS)
```

grade Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.

language English

aims / competences to be developed

Description, assessment, development and application of security mechanisms, techniques and tools.

content

- Basic Cryptography,
- Specification and verification of security protocols,
- Security policies: access control, information flow analysis,
- Network security,
- Media security,
- Security engineering

literature & reading

Will be announced on the course website

Stochastik I

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**jedes Sommersemester**	**1 Semester**	**6**	**9**

responsible Prof. Dr. Henryk Zähle
Prof. Dr. Christian Bender

lecturers Dozent/inn/en der Mathematik

entrance requirements none

assessments / exams Regelmäßige, aktive Teilnahme an der Vorlesung und an den begleitenden Übungen; Abschlussprüfung

course types / weekly hours

```
  4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS
```

total workload

```
   60 h Vorlesung
+  30 h Übungen
+ 180 h Eigenstudium
        (Vor- und Nachbereitung, Bearbeitung Übungsaufgaben)
= 270 h (= 9 ECTS)
```

grade Durch Klausur(en) und mündliche Prüfung. Der Modus wird zu Beginn der Vorlesung bekannt gegeben.

language Deutsch

aims / competences to be developed

content

- Maß- und Integrationstheorie
- Allgemeine Wahrscheinlichkeitsräume
- Zufallsvariablen und deren Verteilungen
- Bedingen auf Ereignisse
- Unabhängigkeit
- Erwartungswert, Varianz, Kovarianz, Korrelation
- Charakterisieren von Verteilungen auf euklidischen Räumen (Verteilungsfunktion, erzeugende Funktionen)
- Summen unabhängiger Zufallsvariablen
- Konvergenzbegriffe für Folgen von Wahrscheinlichkeitsmaßen und Folgen von Zufallsvariablen
- Grenzwertsätze für Summen unabhängiger reellwertiger Zufallsvariablen (Gesetze der großen Zahlen, zentraler Grenzwertsatz)
- Multivariate Normalverteilung, multivariater zentraler Grenzwertsatz

literature & reading

Bekanntgabe jeweils vor der Vorlesung auf der Vorlesungsseite im Internet.

additional information

Methoden: Information durch Vorlesung; Vertiefung durch Eigentätigkeit (Nacharbeit, Übungen).

Anmeldung: Bekanntgabe jeweils rechtzeitig vor Semesterbeginn durch Aushang und im Internet.

Stochastik II

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**jedes Wintersemester**	**1 Semester**	**6**	**9**

responsible Prof. Dr. Henryk Zähle
Prof. Dr. Christian Bender

lecturers Dozent/inn/en der Mathematik

entrance requirements none

assessments / exams Regelmäßige, aktive Teilnahme an der Vorlesung und an den begleitenden Übungen; Abschlussprüfung

course types / weekly hours

```
  4 SWS Vorlesung
+ 2 SWS Übung
= 6 SWS
```

total workload

```
   60 h Vorlesung
+  30 h Übungen
+ 180 h Eigenstudium
        (Vor- und Nachbereitung, Bearbeitung Übungsaufgaben)
= 270 h (= 9 ECTS)
```

grade Durch Klausur(en) und mündliche Prüfung. Der Modus wird zu Beginn der Vorlesung bekannt gegeben.

language Deutsch

aims / competences to be developed

content

- Bedingen auf Sigma-Algebren
- Grundlagen stochastischer Prozesse
- Poisson-Prozess
- Brown'sche Bewegung
- Martingaleigenschaft
- Markoveigenschaft

literature & reading

Bekanntgabe jeweils vor der Vorlesung auf der Vorlesungsseite im Internet.

additional information

Methoden: Information durch Vorlesung; Vertiefung durch Eigentätigkeit (Nacharbeit, Übungen).

Anmeldung: Bekanntgabe jeweils rechtzeitig vor Semesterbeginn durch Aushang und im Internet.

Trusted AI Planning

TAIP

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**4**	**6**

responsible Prof. Dr. Jörg Hoffmann

lecturers Prof. Dr. Jörg Hoffmann

entrance requirements *Programming 1*, *Programming 2*, *Fundamentals of Data Structures and Algorithms*, and *Elements of Machine Learning* or other courses in machine learning are recommended. The *Artificial Intelligence* core course provides useful background but is not necessary.

assessments / exams

- Regular attendance of classes and tutorials
- Solving of weekly assignments
- Passing the final written exam
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

course types / weekly hours

```
  2 h lectures
+ 2 h tutorial
= 4 h (weekly)
```

total workload

```
   60 h of classes
+ 120 h private study
= 180 h (= 6 ECTS)
```

grade Will be determined from the performance in exams. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

Knowledge about methods for learning, verifying and testing action policies in AI Planning; understanding of algorithmic techniques enabling these methods.

content

- Introduction to basic AI concepts needed in the course
- Partial-order reduction
- Dominance pruning
- SAT-based planning
- ASNet action policies
- Safety verification of neural action policies, basic methods
- Safety verification of neural action policies: policy predicate abstraction
- Testing methods for learned action policies, deterministic and probabilistic settings

literature & reading

There is no text book covering the course topics. Links to relevant publications and other material where available will be provided on the slides

additional information

This module was formerly also known as *AI Planning*.

Module Category 4

Seminar DSAI

Seminar

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1-3	**4**	**jedes Semester**	**1 Semester**	**2**	**7**

responsible Studiendekan der Fakultät Mathematik und Informatik
Studienbeauftragter der Informatik

lecturers Dozent/inn/en der Fachrichtung

entrance requirements Grundlegende Kenntnisse im jeweiligen Teilbereich des Studienganges.

assessments / exams

- Thematischer Vortrag mit anschließender Diskussion
- Aktive Teilnahme an der Diskussion
- Gegebenenfalls schriftliche Ausarbeitung oder Projekt

course types / weekly hours `2 SWS Seminar`

total workload

```
   30 h Präsenzstudium
+ 180 h Eigenstudium
= 210 h (= 7 ECTS)
```

grade Wird aus den Leistungen im Vortrag und der schriftlichen Ausarbeitung und/oder dem Seminarprojekt ermittelt. Die genauen Modalitäten werden von dem/der jeweiligen Dozenten/in bekannt gegeben.

language Deutsch oder Englisch

aims / competences to be developed

Die Studierenden haben am Ende der Veranstaltung vor allem ein tiefes Verständnis aktueller oder fundamentaler Aspekte eines spezifischen Teilbereiches der Informatik erlangt.

Sie haben weitere Kompetenz im eigenständigen wissenschaftlichen Recherchieren, Einordnen, Zusammenfassen, Diskutieren, Kritisieren und Präsentieren von wissenschaftlichen Erkenntnissen gewonnen.

content

Weitgehend selbstständiges Erarbeiten des Seminarthemas:

- Lesen und Verstehen wissenschaftlicher Arbeiten
- Analyse und Bewertung wissenschaftlicher Aufsätze
- Diskutieren der Arbeiten in der Gruppe
- Analysieren, Zusammenfassen und Wiedergeben des spezifischen Themas
- Erarbeiten gemeinsamer Standards für wissenschaftliches Arbeit
- Präsentationstechnik

Spezifische Vertiefung in Bezug auf das individuelle Thema des Seminars.

Der typische Ablauf eines Seminars ist üblicherweise wie folgt:

- Vorbereitende Gespräche zur Themenauswahl
- Regelmäßige Treffen mit Diskussion ausgewählter Beiträge
- ggf. Bearbeitung eines themenbegleitenden Projekts
- Vortrag und ggf. Ausarbeitung zu einem der Beiträge

literature & reading

Material wird dem Thema entsprechend ausgewählt.

additional information

Die jeweils zur Verfügung stehenden Seminare werden vor Beginn des Semesters angekündigt und unterscheiden sich je nach Studiengang.

Module Category 5

Master-Seminar und -Arbeit

Master Seminar

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
3	4	every semester	1 semester	2	12

responsible	Dean of Studies of the Faculty of Mathematics and Computer Science Study representative of computer science
lecturers	Professors of the department
entrance requirements	Acquisition of at least 30 CP
assessments / exams	• Preparation of the relevant scientific literature • Written elaboration of the topic of the master thesis • Presentation about the planned topic with subsequent discussion • Active participation in the discussion
course types / weekly hours	2 h seminar (weekly)
total workload	30 h seminar + 40 h contact with supervisor + 290 h private study = 360 h (= 12 ECTS)
grade	graded
language	English or German

aims / competences to be developed

The Master seminar sets the ground for carrying out independent research within the context of an appropriately demanding research area. This area provides sufficient room for developing own scientific ideas.

At the end of the Master seminar, the basics ingredients needed to embark on a successful Master thesis project have been explored and discussed with peers, and the main scientific solution techniques are established.

The Master seminar thus prepares the topic of the Master thesis. It does so while deepening the students' capabilities to perform a scientific discourse. These capabilities are practiced by active participation in a reading group. This reading group explores and discusses scientifically demanding topics of a coherent subject area.

content

The methods of computer science are systematically applied, on the basis of the "state-of-the-art".

literature & reading

Scientific articles corresponding to the topic area in close consultation with the lecturer.

Master Thesis

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
4	**4**	**every semester**	**6 months**	**-**	**30**

responsible Dean of Studies of the Faculty of Mathematics and Computer Science
Study representative of computer science

lecturers Professors of the department

entrance requirements Successful completion of the *Master Seminar*

assessments / exams Written elaboration in form of a scientific paper. It describes the scientific findings as well as the way leading to these findings. It contains justifications for decisions regarding chosen methods for the thesis and discarded alternatives. The student's own substantial contribution to the achieved results has to be evident. In addition, the student presents his work in a colloquium, in which the scientific quality and the scientific independence of his achievements are evaluated.

course types / weekly hours none

total workload

```
   50 h contact with supervisor
+ 850 h private study
= 900 h (= 30 ECTS)
```

grade Grading of the Master Thesis

language English or German

aims / competences to be developed

In the master thesis the student demonstrates his ability to perform independent scientific work focusing on an adequately challenging topic prepared in the master seminar.

content

In the master thesis the student demonstrates his ability to perform independent scientific work focusing on an adequately challenging topic prepared in the master seminar.

literature & reading

According to the topic