UNIVERSITÄT
DES
SAARLANDES

Faculty of Mathematics and Computer Science

Module Descriptions

Medieninformatik MSc

17th December 2024

List of module categories and modules

Module Category 1

Stammvorlesungen

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.

Artificial Intelligence AI

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	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	**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.

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.

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*.

Database Systems DBS

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	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.

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.

Image Processing and Computer Vision IPCV

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	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.

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.

Machine Learning ML

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	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.

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.

Security

Sec

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	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

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

Vertiefungsvorlesungen

Audio-Visual Communication and Networks AVCN

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

responsible Prof. Dr.-Ing. Thorsten Herfet

lecturers Prof. Dr.-Ing. Thorsten Herfet

entrance requirements Solid foundation of mathematics (differential and integral calculus) and probability theory. The course will build on the mathematical concepts and tools taught in TC I while trying to enable everyone to follow and to fill gaps by an accelerated study of the accompanying literature. *Signals and Systems* as well as *Digital Transmission and Signal Processing (TC I)* are strongly recommended but not required.

assessments / exams Regular attendance of classes and tutorials Passing the final exam
Oral exam directly succeeding the course. Eligibility: Weekly excersises / 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

AVCN will deepen the students' knowledge on modern communications systems and will focus on wireless systems.

Since from a telecommunications perspective the combination of audio/visual data – meaning inherently high data rate and putting high requirements on the realtime capabilities of the underlying network – and wireless transmission – that is unreliable and highly dynamic with respect to the channel characteristics and its capacity – is the most demanding application domain.

content

As the basic principle the course will study and introduce the building blocks of wireless communication systems. Multiple access schemes like TDMA, FDMA, CDMA and SDMA are introduced, antennas and propagation incl. link budget calculations are dealt with and more advanced channel models like MIMO are investigated. Modulation and error correction technologies presented in Telecommunications I will be expanded by e.g. turbo coding and receiver architectures like RAKE and BLAST will be introduced. A noticeable portion of the lecture will present existing and future wireless networks and their extensions for audio/visual data. Examples include 802.11n and the terrestrial DVB system (DVB-T2).

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 II*.

Automata, Games and Verification AGV

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

responsible	Prof. Bernd Finkbeiner, Ph.D
lecturers	Prof. Bernd Finkbeiner, Ph.D
entrance requirements	none
assessments / exams	• Regular attendance of classes and tutorial • 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	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 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 gain a deep understanding of the automata-theoretic background of automated verification and program synthesis.

content

The theory of automata over infinite objects provides a succinct, expressive and formal framework for reasoning about reactive systems, such as communication protocols and control systems. Reactive systems are characterized by their nonterminating behaviour and persistent interaction with their environment.

In this course we study the main ingredients of this elegant theory, and its application to automatic verification (model checking) and program synthesis.

- Automata over infinite words and trees (omega-automata)
- Infinite two-person games
- Logical systems for the specification of nonterminating behavior
- Transformation of automata according to logical operations

literature & reading

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

Automated Debugging

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

responsible	Prof. Dr. Andreas Zeller
lecturers	Prof. Dr. Andreas Zeller
entrance requirements	*Programmierung 1*, *Programmierung 2* and *Softwarepraktikum*
assessments / exams	Projects and mini-tests

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	The module is passed in its entirety if the examination performance has been passed.
language	English

aims / competences to be developed

Finding and fixing software bugs can involve lots of effort. This course addresses this problem by automating software debugging, specifically identifying failure causes, locating bugs, and fixing them. Students learn the basics of systematic debugging, and explore tools and techniques for automated debugging.

content

- Tracking Problems
- The Scientific Method
- Cause-Effect Chains
- Building a Debugger
- Tracking Inputs
- Assertions and Sanitizers
- Detecting Anomalies
- Statistical Fault Localization
- Generating Tests
- Reducing Failure-Inducing Inputs
- Mining Software Archives
- Fixing the Defect
- Repairing Bugs Automatically
- Managing Bugs

literature & reading

The teaching material consists of text, Python code, and Jupyter Notebooks from the textbook "The Debugging Book" (https://www.debuggingbook.org/), also in English.

Computer Architecture 2 CAr2

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

responsible	Prof. Dr. W.-J. Paul
lecturers	Prof. Dr. W.-J. Paul
entrance requirements	Related core lecture *Computer Architecture*
assessments / exams	**Studying:** Students should to listen to the lectures, read the lecture notes afterwards and understand them. They should solve the exercises alone or in groups. Students must present and explain their solutions during the tutorials. **Exams:** students who have solved 50 % of all exercises are allowed to participate in an oral 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 this lecture students know how to design IEEE compatible floationg point units and some form of parallel computer system.

content

General comment: constructions are usually presented together with correctness proofs; Below you find the 2005/2006 Version of this lecture

- Basics of Floating Point Computation
  - IEEE standard
  - Theory of rounding
- FPU construction
  - Add/subtract unit
  - Multiply/divide unit
  - Rounding
- Automotive systems hardware
  - Serial interfaces
  - Clock Synchronization
  - FlexRay like Interfaces
  - Electronic control units
- Automotive systems software
  - An OSEKTime like programming model

- An OSEKTime like real time operating system
- Drivers
- Worst Case Execution Time
- Pervasive Correctness proof

literature & reading

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

Correspondence Problems in Computer Vision CoPCV

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
3	4	occasional	1 semester	4	6

responsible Prof. Dr. Joachim Weickert

lecturers Dr. Pascal Peter

entrance requirements Undergraduate mathematics (e.g. "Mathematik für Informatiker I-III") is required, as well as elementary C knowledge (for the programming assignments). Knowledge in image processing or differential equations is useful.

assessments / exams
- Regular attendance of lecture and tutorial
- Written or oral exam and the end of the course

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 performance in exams. The exact modalities will be announced at the beginning of the module.

language English

aims / competences to be developed

Correspondence problems are a central topic in computer vision. Thereby, one is interested in identifying and matching corresponding features in different images/views of the same scene. Typical corresondence problems are the estimation of motion information from consecutive frames of an image sequence (optic flow), the reconstruction of a 3-D scene from a stereo image pair and the registration of medical image data from different modalities (e.g. CT and MRT). Central part of this lecture is the discussion of the most important correspondence problems as well as the modelling of suitable algorithms for solving them.

content

```
1.  Introduction and Overview
2.  General Matching Concepts
    2.1 Block Matching
    2.2 Correlation Techniques
    2.3 Interest Points
    2.4 Feature-Based Methods
3.  Optic Flow I
    3.1 Local Differential Methods
    3.2 Parameterisation Models
4.  Optic Flow II
    4.1 Global Differential Methods
    4.2 Horn and Schunck
5.  Optic Flow III
    5.1 Advanced Constancy Assumptions
    5.2 Large Motion
```

```
6.  Optic Flow IV
    6.1 Robust Data Terms
    6.2 Discontinuity-Preserving Smoothness Terms
7.  Optic Flow V
    7.1 High Accuracy Methods
    7.2 SOR and Lienar Multigrid
8.  Stereo Matching I
    8.1 Projective Geometry
    8.2 Epipolar Geometry
9.  Stereo Matching II
    9.1 Estimation of the Fundamental Matrix
10. Stereo Matching III
    10.1 Correlation Methods
    10.2 Variational Approaches
    10.3 Graph Cuts
11. Medical Image Registration
    11.1 Mutual Information
    11.2 Elastic and Curvature Based Registration
    11.3 Landmarks
12. Particle Image Velocimetry
    12.1 Div-Curl-Regularisation
    12.2 Incompressible Navier Stokes Prior
```

literature & reading

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

Introduction to Image Acquisition Methods IIAM

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
3	4	at least every two years	1 semester	2	4

responsible Prof. Dr. Joachim Weickert

lecturers N.N.

entrance requirements Related core lecture *Computer Vision*

assessments / exams
- Written or oral exam at end of course
- 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 (weekly)
```

total workload

```
   30 h of classes
+  90 h private study
= 120 h (= 4 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 course is designed as a supplement for image processing lectures, to be attended before, after or parallel to them.

Participants shall understand

- what are digital images
- how they are acquired
- what they encode and what they mean
- which limitations are introduced by the image acquisition.

This knowledge will be helpful in selecting adequate methods for processing image data arising from different methods.

content

A broad variety of image acquisition methods is described, including imaging by virtually all sorts of electromagnetic waves, acoustic imaging, magnetic resonance imaging and more. While medical imaging methods play an important role, the overview is not limited to them.

Starting from physical foundations, description of each image acquisition method extends via aspects of technical realisation to mathematical modelling and representation of the data.

literature & reading

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

Module Category 3

Seminare

Seminar

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
1	**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 4

HBKsaar

MAD-Projekt MAD-P

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
		jedes Semester	**1 Semester**	**8**	**8**

responsible	Dozent/inn/en der Hochschule der Bildenden Künste (HBKsaar)
lecturers	Dozent/inn/en der Hochschule der Bildenden Künste (HBKsaar)
entrance requirements	*Grundlagen Media Art & Design* (empfohlen)
assessments / exams	Projektarbeit, Projektdokumentation, Projektpräsentation
course types / weekly hours	8 SWS Projektarbeit

total workload	`  120 h Projektarbeit`
	`+ 120 h Vor- und Nachbereitung`
	`        (Projektrecherche und -dokumentation)`
	`= 240 h (= 8 ECTS)`

grade	Das Modul ist insgesamt bestanden, wenn die Prüfungsleistung bestanden wurde. (benotet)
language	Deutsch oder Englisch

aims / competences to be developed

Die konkreten Lernziele der einzelnen Veranstaltungen werden jährlich von der Hochschule der Bildenden Künste (HBKsaar) aktualisiert und bekannt gegeben. Kurzüberblick der vermittelten Kompetenzen:

- Reflexion diverser Medien, Materialien und Methoden
- Kritische Reflexion medialer Bildwelten und von Phänomenen der Alltagsästhetik sowie deren Darstellung in rhetorischen Grundformen
- Bewirken ästhetischer Entscheidungen in Gestaltungsprozessen sowie deren begründete Reflexion
- Entwurf, Vorbereitung und Durchführung eines eigenständig bearbeiteten Projekts
- Projektdokumentation, die das Projekt und die eigene Leistung dokumentiert
- Präsentation der eigenen Arbeit in einem Ateliergespräch oder einer Ausstellung

content

Die konkreten Inhalte orientieren sich an den angebotenen Projektarbeiten und werden jährlich von der Hochschule der Bildenden Künste (HBKsaar) aktualisiert und bekannt gegeben. Die Themen spannen einen attraktiven Bogen von aktueller Forschung zu anspruchsvollen Problemen der industriellen Praxis.

literature & reading

Die Literatur zum Modul kann englisch- und/oder deutschsprachig sein und wird zu Beginn der Veranstaltung bekannt gegeben.

additional information

Die zugehörigen Veranstaltungen werden im Vorlesungszeichnis der HBKsaar als *Atelierprojekt kurz* bezeichnet.

Module Category 5

Praktikumsphase

Berufspraktikum & Praktikumsseminar BB&PS

st. semester	std. st. sem.	cycle	duration	SWS	ECTS
2	4	jedes Sommersemester	1 Semester	PRAKTIKUM	20+5

responsible	Prof. Dr. Antonio Krüger
lecturers	Prof. Dr. Antonio Krüger
entrance requirements	Genehmigung des Themengebietes und Inhalts des Berufspraktikums durch die Fachstudienberatung oder den Prüfungsausschuss nach *§6 Praktikumsphase* der Studienordnung.
assessments / exams	Projektarbeit, Praktikumsbericht und Kolloquium.
course types / weekly hours	**Modul: Berufspraktikum:** Projektarbeit in der Industrie oder der Forschung nach *§6 Praktikumsphase* der Studienordnung. **Modul: Praktikumsseminar:** Vortrag über das durchgeführte Berufspraktikum, Praktikumsbericht, und Teilnahme am Kolloquium.

total workload	`  600 h Berufspraktikum`
	`+ 150 h Praktikumsseminar`
	`= 750 h (= 25 ECTS)`

grade	Das Modul ist insgesamt bestanden, wenn die Prüfungsleistung bestanden wurde (Berufspraktikum: unbenotet, Praktikumsseminar: benotet). Modulnote aus der Beurteilung des abschließenden Kolloquiums.
language	Deutsch oder Englisch

aims / competences to be developed

Das Berufspraktikum ist ein Praktikum zu einem Themengebiet der Medieninformatik oder einem verwandten Fachgebiet wie bspw. Informatik, das in der Industrie oder der Forschung durchgeführt wird.

Die Praktikumsphase soll die Studierenden befähigen, wissenschaftliche oder industrielle Aufgabenstellungen aus dem Bereich der Medieninformatik oder einem verwandten Fachgebiet wie bspw. Informatik ganzheitlich zu bearbeiten und die Ergebnisse in nachvollziehbarer Form zu dokumentieren, auszuarbeiten und in einem Praktikumsseminar zu präsentieren.

Ziel des Berufspraktikums ist die Anwendung fachspezifischen Wissens zur Bearbeitung realitätsnaher Aufgabenstellungen, die Fähigkeit zur Selbstorganisation und die Eingliederung in ein bestehendes Projektteam.

- Umsetzung und Anwendung der Lehrinhalte des Studiengangs
- Zielorientiertes Arbeiten in einem Team unter Randbedingungen der Industrie
- Fähigkeit zur Präsentation und Verteidigung der Ergebnisse

content

- Bearbeitung eines Themengebietes der Medieninformatik in einem industriellen Umfeld
- Aufbereitung und Anwendung von Fachwissen und Methoden
- Erarbeitung, Erprobung und Bewertung von Lösungsansätzen
- selbstorganisierte Teamarbeit und Rollenverteilung
- Dokumentation des Praktikumsverlaufs und der Projektarbeit
- Präsentation der Arbeiten und Ergebnisse in einem Vortrag mit abschließendem Kolloquium

literature & reading

Material wird dem Thema entsprechend ausgewählt.

Module Category 6

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