UNIVERSITÄT DES SAARLANDES

Module Descriptions

Master Program Computer Science

(01.10.2016)

Compulsory courses:
A compulsory course must be taken to gain the relevant qualification.

Mandatory Elective Courses:
Mandatory elective courses give students a restricted choice. Students must complete a certain number of mandatory elective courses from a set of options to fulfil a certain category given by the examination regulations.

Elective Courses:
Not all courses chosen need necessarily come from the degree program being studied. Some courses offered by other faculties in the UdS can be used to contribute credit points towards the final degree.

Rs.-Sem.	Modul	CP	SWS
Core Courses (Mandatory Elective Courses)			
1-3	Operating Systems	9	6
1-3	Computer Graphics	9	6
1-3	Database Systems	9	6
1-3	Embedded Systems	9	6
1-3	Information Retrieval and Data Mining	9	6
1-3	Artificial Intelligence	9	6
1-3	Computer Architecture	9	6
1-3	Security	9	6
1-3	Software Engineering	9	6
1-3	Compiler Construction	9	6
1-3	Automated Reasoning	9	6
1-3	Image Processing and Computer Vision	9	6
1-3	Computer Algebra	9	6
1-3	Algorithms and Data Structures	9	6
1-3	Introduction to Computational Logic	9	6
1-3	Geometric Modelling	9	6
1-3	Complexity Theory	9	6
1-3	Cryptography	9	6
1-3	Optimization	9	6
1-3	Semantics	9	6
1-3	Verification	9	6
1-3	Telecommunications I	9	6

Rs.-Sem.	Modul	CP	SWS
1-3	Machine Learning	9	6
1-3	Distributed Systems	9	6
1-3	Data Networks	9	6
	Advanced Courses (Elective Courses)		
2-3	Computer Architecture 2	9	6
2-3	Telecommunications II	9	6
2-3	Automata, Games and Verification	6	4
2-3	Automated Debugging	6	4
2-3	Computer Graphics II	9	6
2-3	Differential Equations in Image Processing and Computer Vision	9	6
2-3	Introduction to Image Acquisition Methods	4	2
2-3	Correspondence Problems in Computer Vision	6	4
2-3	Future Media Internet	9	6
2-3	Automatic Planning	9	4
1-3	Seminar	7	3
4	Master-Seminar	12	5
4	Master Thesis	30	
1-3	Tutor	4	2
1-3	Soft Skill	4 o. 6	2 o. 4
1 - 3	Language Courses	4 o. 6	2 o. 4
	Praktikum zum Informationsmanagement	6 o. 9	4 o. 6

Operating Systems, Core Course					**CS 551 / OS**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Peter Druschel, Ph.D.
Lecturer	Prof. Peter Druschel, Ph.D. Björn Brandenburg, Ph.D.
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Computer Graphics, Core Course					**CS 552 / CG**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Philipp Slusallek
Lecturer	Prof. Dr. Philipp Slusallek
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective)
Entrance requirements	For graduate students: none
Assessment / Exams	• Sucessful completion of at least 50% of the exercises • Sucessful participation in rendering competition • Final written exam Final grade determined by result of the exam and the rendering competition A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

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 focuses on image synthesis or rendering. After introducing of physical background and the representations used in graphics it discusses the two basic algorithms for image synthesis: ray tracing and rasterization. In this context we present related topics like texturing, shading, aliasing, sampling, and many more. As part of the practical exercises the students incrementally build their own ray tracing system or hardware-based visualization application. A final rendering competition allows students to implement their favorite advanced algorithm and and use it in a high-quality rendering.

Content

- Fundamentals of digital image synthesis
    - Phyical laws of light transport
    - Human visual system and perception
    - Colors and Tone-Mapping
    - Signal processing and anti-aliasing
    - Materials and reflection models
    - Geometric modeling
    - Camera models
- Ray Tracing
    - Recursive ray tracing algorithm
    - Spatial index structures
    - Sampling approaches
    - Parallel and distributed algorithms
- Rasterization and Graphics Hardware
    - Homogeneous coordinates, transformations
    - Hardware architectures
    - Rendering pipeline
    - Shader programming and languages
    - OpenGL

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Database Systems, Core Course					**CS 553 / DBS**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer Prof. Dr. Jens Dittrich

Lecturer Prof. Dr. Jens Dittrich

Level of the unit / mandatory or not Graduate course / Mandatory Elective

Entrance requirements

especially Saarland University CS department's undergraduate lecture "Informationssysteme", "Prog 1", "Prog 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

Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly; this class may be run as a flipped classroom, i.e. 2 hours 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”) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade	Will be determined based on project, midterm and best of endterm and reexam.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Embedded Systems, Core Course					**CS 650 /ES**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer Prof. Bernd Finkbeiner, Ph.D

Lecturer Prof. Bernd Finkbeiner, Ph.D

Level of the unit / mandatory or not
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Entrance requirements

Assessment / Exams
- Written exam at the end of the course.
- Demonstration of the implemented system.
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Course Typ / weekly hours
Lecture 4 h (weekly)
Tutorial 2 h (weekly)
The course is accompanied by a laboratory project, in which a non-trivial embedded system has to be realized.

Total workload 270 h = 90 h classes and 180 h private study

Grade of the module 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.

Aims / Competences to be developed

The students should learn methods for the design, the implementation, and the validation of safety-critical embedded systems.

Content

Embedded Computer Systems are components of a technical system, e.g. an air plane, a car, a household machine, a production facility. They control some part of this system, often called the plant, e.g. the airbag controller in a car controls one or several airbags. Controlling means obtaining sensor values and computing values of actuator signals and sending them.
Most software taught in programming courses is transformational, i.e. it is started on some input, computes the corresponding output and terminates. Embedded software is reactive, i.e. it is continuously active waiting for signals from the plant and issuing signals to the plant.
Many embedded systems control safety-critical systems, i.e. malfunctioning of the system will in general cause severe damage. In addition, many have to satisfy real-time requirements, i.e. their reactions to input have to be produced within fixed deadlines.
According to recent statistics, more than 99% of all processors are embedded. Processors in the ubiquitous PC are a negligible minority. Embedded systems have a great economical impact as most innovations in domains like avionics, automotive are connected to advances in computer control. On the other hand, failures in the design of such systems may have disastrous consequences for the functioning of the overall system. Therefore, formal specification techniques and automatic synthesis of software are used more than in other domains.
The course will cover most aspects of the design and implementation of embedded systems, e.g. specification mechanisms, embedded hardware, operating systems, scheduling, validation methods.

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Information Retrieval and Data Mining, Core Course					**CS 555 / IRDM**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer Prof. Dr. Gerhard Weikum

Lecturer Prof. Dr. Gerhard Weikum

Level of the unit / mandatory or not
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

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

Assessment / 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 Typ / weekly hours
Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Total workload
270 h = 90 h of classes and 180 h private study

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

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 contents 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.

Additional Information

Teaching language: English

Literature:
will be announced on the course web site.

Artificial Intelligence, Core Course					**CS 556 / AI**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Jörg Hoffmann
Lecturer	Prof. Dr. Jörg Hoffmann, Prof. Dr. Wolfgang Wahlster
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 30 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Wird aus Leistungen in Klausuren ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Aims / Competences to be developed

Knowledge about basic methods in Artificial Intelligence

Content

Problem-solving:
- Uninformed- and informed search procedures
- Adversarial search

Knowledge and reasoning:
- Propositional logic
- SAT
- First-order logic, Inference in first-order logic
- Knowledge representation, Semantic Web
- Default logic, rule-based mechanisms

Planning:
- STRIPS formalism and complexity
- Delete relaxation heuristics

Probabilistic reasoning:
- Basic probabilistic methods
- Bayesian networks

Additional Information

Teaching language: English

Literature:
Russel & Norvig „Artificial Intelligence: A Modern Approach“
Additional optionally will be announced on the course website

Computer Architecture, Core Course					**CS 558 / CAR**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. W.-J. Paul
Lecturer	Prof. Dr. W.-J. Paul
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	Studying: Students should 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 at the end of the semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

Aims / Competences to be developed

After attending this lecture students know how to design pipelined processors with interrupt mechanisms, caches and MMUs. Given a benchmark they know how to analyse, whether a change makes the processor more or less cost effective.

Content

General comment: constructions are usually presented together with correctness proofs

- Complexity of Architectures
  - Hardware cost and cycle time
  - Compilers and benchmarks
- Circuits
  - Elementary computer arithmetic

    - Fast adders
    - Fast multipliers
- Sequential processor design
    - DLX instruction set
    - Processor design
- Pipelining
    - Elementary pipelining
    - Forwarding
    - Hardware-Interlock
- Interrupt mechanisms
    - Extension of the instruction set
    - Interrupt service routines
    - hardware construction
- Caches
    - Specification including consistency between instruction and data cache
    - Cache policies
    - Bus protocol
    - Hardware construction (k-way set associative cache, LRU replacement, realisation of bus protocols by automat)
- Operating System Support
    - Virtual and Physical machines
    - Address translation
    - Memory management unit (MMU) construction
    - Virtual memory simulation

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Security, Core Course					**CS 559 / SEC**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer — Prof. Dr. Michael Backes

Lecturer — Prof. Dr. Michael Backes

Level of the unit / mandatory or not — Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Entrance requirements — For graduate students: none

Assessment / Exams

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

Course Typ / weekly hours — Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Total workload — 270 h = 90 h of classes and 180 h private study

Grade of the module — 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.

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

Additional Information
Teaching language: English

Literature:
Will be announced on the course website

Software Engineering, Core Course					**CS 560 / SE**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Andreas Zeller
Lecturer	Prof. Dr. Andreas Zeller
Level of the unit / mandatory or not	Graduate course / Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	• Successful project completion (including deliverables such as requirements, design, implementation) • Successful project demonstration • Regular attendance of classes • Passing the final exam
Course Typ / weekly hours	Lecture 2 h (weekly) Project 4 h (weekly) Project work in teams of 4–7 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

Aims / Competences to be developed

The students know and apply modern software development techniques.

They are aware of systematic elicitation of requirements and how to document them.

They are aware of advanced quality assurance techniques such as test coverage, program analysis, and verification and know about the appropriate standards.

They know modern paradigms of programming and design, and know when to use them.

They know the standards of project management and project organization and can assess the state of given projects as well as suggest consequences to reach specific targets.

They apply these techniques in a project in small teams.

Lecture Contents

- Software Processes (Testing process, ISO 9000, maturity model, extreme programming)
- Modeling and design (requirements engineering, formal specification, proofs, model checking)
- Programming paradigms (aspect-oriented, generative, and component-based programming)
- Validation (Testing, Reliability assessment, tools)
- Software maintenance (configuration management, reengineering, restructuring)
- Project skills (organization, structure, estimations)
- Human resources (communication, assessment)
  Controlling (metrics, change requests, risk and quality managament)

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Compiler Construction, Core Course					**CS 561 / CC**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Sebastian Hack
Lecturer	Prof. Dr. Sebastian Hack
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

Aims / Competences to be developed

The students learn, how a source program is lexically, syntactically, and semantically analyzed, and how they re 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.

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Automated Reasoning, Core Course					**CS 571 / AR**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Christoph Weidenbach
Lecturer	Prof. Dr. Christoph Weidenbach
Level of the unit / mandatory or not	Graduate course / Mandatory Elective
Entrance requirements	CS 575 ICL
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Image Processing and Computer Vision, Core Course					**CS 572 / IPCV**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Joachim Weickert
Lecturer	Prof. Dr. Joachim Weickert
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	• Regular attendance of classes and tutorials. • At least 50% of all possible points from the weekly assignments have to be gained to qualify fort he final exam. • Passing the final exam • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

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

1. Basics
   1.1 Image Types and Discretisation
   1.2 Degradations in Digital Images
2. Image Transformations
   2.1 Fourier Transform
   2.2 Image Pyramids
   2.3 Wavelet Transform
3. Colour Perception and Colour Spaces

4. Image Enhancement
   4.1 Point Operations
   4.2 Linear Filtering
   4.3 Wavelet Shrinkage, Median Filtering, M-Smoothers
   4.4 Mathematical Morphology
   4.5 Diffusion Filtering
   4.6 Variational Methods
   4.7 Deblurring
5. Feature Extraction
   5.1 Edges
   5.2 Corners
   5.3 Lines and Circles
6. Texture Analysis
7. Segmentation
   7.1 Classical Methods
   7.2 Variational Methods
8. Image Sequence Analysis
   8.1 Local Methods
   8.2 Variational Methods
9. 3-D Reconstruction
   9.1 Camera Geometry
   9.2 Stereo
   9.3 Shape-from-Shading
10. Object Recognition
   10.1 Eigenspace Methods
   10.2 Moment Invariances

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Computer Algebra, Core Course					**CS 573 / CA**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Frank-Olaf Schreyer
Lecturer	Prof. Dr. Frank-Olaf Schreyer
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	• Regular attendance of classes and tutorials • Solving the exercises, passing the midterm and the final exam. • Grade: 20% exercises, 30% midterm, 50% final exam.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Algorithms and Data Structures, Core Course					**CS 574 / A&D**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Kurt Mehlhorn
Lecturer	Prof. Dr. Kurt Mehlhorn, Prof. Dr. Raimund Seidel
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: C, C++, Java
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

Aims / Competences to be developed

The students know standard algorithms for typical problems in the areas graphs, computational geometry, strings and optimization. Additionalmore 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.

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Introduction to Computational Logic, Core Course					**CS 575 / ICL**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer Prof. Dr. Gert Smolka

Lecturer Prof. Dr. Gert Smolka

Level of the unit / mandatory or not
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Entrance requirements

Assessment / Exams
- Regular attendance of classes and tutorials.
- Passing the midterm and the final exam.

Course Typ / weekly hours
Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Total workload 270 h = 90 h of classes and 180 h private study

Grade of the module 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.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Geometric Modeling, Core Course					**CS 576 / GM**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Hans-Peter Seidel
Lecturer	Prof. Dr. Hans-Peter Seidel, Prof. Dr. Philipp Slusallek
Level of the unit / mandatory or not	Graduate course / Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students (theory) Practical assignments in groups of 3 students (practice) Tutorials consists of a mix of theoretical + practical assignments.
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be based on the performance in exams, exercises and practical tasks. The detailed terms will be announced by the module coordinator.

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 Modeling
- Mesh processing
- Approximation of differential operators
- Shape Analysis and Geometry Processing

Additional Information

Teaching language:English

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

Complexity Theory, Core Course					**CS 577 / CT**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Markus Bläser
Lecturer	Prof. Dr. Markus Bläser, Prof. Dr. Raimund Seidel
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	none undergraduate course on theory of computation (e.g. "Grundzüge der Theoretischen Informatik“) is highly recommend.
Assessment / Exams	• Regular attendance of classes and tutorials • assignments • exams (written or oral)
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of about 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be calculated from the results in the assignments and/or exams, as announced by the lecturer at the beginning of the course

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.

Additional Information

Teaching language: English

Literature:
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

Cryptography, Core Course					**CS 578 / CRY**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Michael Backes
Lecturer	Prof. Dr. Michael Backes
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: Basic knowledge in theoretical computer science required, background knowledge in number theory and complexity theory helpful
Assessment / Exams	• Oral / written exam (depending on the number of students) • A re-exam is normally provided (as written or oral examination).
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Optimization, Core Course					**CS 579 / OPT**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Kurt Mehlhorn
Lecturer	Prof. Dr. Kurt Mehlhorn
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Semantics, Core Course					**CS 580 / SEM**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Gert Smolka
Lecturer	Prof. Dr. Gert Smolka
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: core lecture Introduction to Computational Logic
Assessment / Exams	• Regular attendance of classes and tutorials. • Passing the midterm and the final exam
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Verification, Core Course					**CS 581 / VERI**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Holger Hermanns
Lecturer	Prof. Dr. Holger Hermanns, Prof. Bernd Finkbeiner, Ph.D
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Telecommunications I, Core Course					**TC I**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr.-Ing. Thorsten Herfet
Lecturer	Prof. Dr.-Ing. Thorsten Herfet
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
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.
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Final exam mark

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 telecommu¬nication systems do incorporate. Sources, sinks, source and channel coding, modulation and multiplexing are the major keywords but we will also deal with dedicated pie¬ces like A/D- and D/A-converters and quantizers in a little bit more depth.
The course will refresh the basic transformations (Fou¬rier, 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 communi¬cation systems.

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Machine Learning, Core Course					**ML**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Matthias Hein
Lecturer	Prof. Dr. Matthias Hein
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
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.
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Determined from the results of the exams, exercises and potential projects. The exact grading modalities are announced at the beginning of the course.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Distributed Systems, Core Course					**DS**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Peter Druschel, Ph.D.
Lecturer	Prof. Peter Druschel, Ph.D. Allen Clement, Ph.D.
Level of the unit / mandatory or not	Graduate course / Mandatory Elective
Entrance requirements	Operating systems or concurrent programming.
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly)
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

Aims / Competences to be developed

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

Content

- Communication: Remote procedure call, distributed objects, event notification, content 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, content distribution networks.
- Data centers. Architecture and infrastructure, distributed programming, energy efficiency.

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Data Networks, Core Course					**CS 554 / DN**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Holger Hermanns
Lecturer	Prof. Dr. Holger Hermanns
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Computer Architecture 2, Advanced Course					CS 650 / CAR2
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. W. J. Paul
Lecturer	Prof. Dr. W. J. Paul
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate / Mandatory Elective
Entrance requirements	Related core lecture Computer Architecture
Assessment / 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 Typ / weekly hours	Lecture 4 h weekly, 50-100 students Tutorials 2 h weekly, up to 20 students
Total workload	270 hours = 90 h classes and 180 h private study
Grade of the module	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.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Telecommunications II, Advanced Course					CS 650 / TC II
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr.-Ing. Thorsten Herfet
Lecturer	Prof. Dr.-Ing. Thorsten Herfet
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
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 "TC I - Digital Transmission and Signal Processing" are strongly recommended but not required. Related core lecture TC I
Assessment / 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 Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 hours = 90 h classes and 180 h private study
Grade of the module	Final Exam Mark

Aims / Competences to be developed

TC II 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).

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Automata, Games and Verification, Advanced Course					CS 650 / AG&V
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	4	6

Responsible Lecturer	Prof. Bernd Finkbeiner, PhD
Lecturer	Prof. Bernd Finkbeiner, PhD
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	
Assessment / 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 Typ / weekly hours	Lecture 2 h (weekly) Tutorial 2 h (weekly)
Total workload	180 h = 60 h classes and 120 h private study
Grade of the module	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.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Automated Debugging, Advanced Course					CS 650 / AutoD
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	4	6

Responsible Lecturer	Prof. Dr. Andreas Zeller
Lecturer	Prof. Dr. Andreas Zeller
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate / Mandatory Elective
Entrance requirements	Programming skills as acquired at the Bachelor level
Assessment / Exams	• Project exercises during the course • 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 Typ / weekly hours	Lecture 2 h (weekly) Tutorial 2 h (weekly)
Total workload	180 h = 60 h classes and 120 h private study
Grade of the module	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.

Aims / Competences to be developed

This is a course about bugs in computer programs, how to reproduce them, how to find them, and how to fix them such that they do not occur anymore. This course teaches a number of techniques that allow you to debug any program in a systematic, and sometimes even elegant way. Moreover, the techniques can widely be automated, which allows you to let your computer do most of the debugging.

Once you understand how debugging works, you won't think about debugging in the same way. Instead of seeing a wild mess of code, you will think about causes and effects, and you will systematically set up and refine hypotheses to track failure causes. Your insights may even make you set up your own automated debugging tool. All of this allows you to spend less time on debugging, which is why you're interested in automated debugging in the first place, right?

Content

Questions this course addresses include:

- How can I reproduce failures faithfully?
- How can I isolate what's relevant for the failure?
- How does the failure come to be?
- How can I fix the program in the best possible way?

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Computer Graphics II, Advanced Course Realistic Image Synthesis					**CS 650 / CGII-RIS**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Philipp Slusallek
Lecturer	Prof. Dr. Philipp Slusallek
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	Related core lecture Computer Graphics
Assessment / Exams	• Theoretical and practical exercises (50% requirement for final exam) • Final oral exam • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly)
Total workload	270 h = 90 h classes and 180 h private study
Grade of the module	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.

Aims / Competences to be developed

At the core of computer graphics is the requirement to render highly realistic and often even physically accurate images of virtual 3D scenes. In this lecture students will learn about physically-based simulation techniques to compute the distribution of light in even complex environment. After this course students should be able to build their own highly realistic but also efficient rendering system.

Content

- Rendering and Radiosity Equation, Finite Elements
- Radiosity
- Monte Carlo Techniques
- Direct Illumination, Importance Sampling
- BRDF, Inversion Methods
- Distribution Ray Tracing and Path Tracing
- Theory of Variance Reduction
- Bidirectional Path Tracing, Instant Radiosity
- Density Estimation Methods
- Photon Mapping
- Rendering of Animations
- Motion Blur, Temporal Filtering
- Interactive Global Illumination
- Hardware Rendering Basics
- Advanced Hardware Rendering
- Measurements of BRDFs and Light Sources
- Relighting
- Tone Mapping, Perception

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Differential Equations in Image Processing and Computer Vision, Advanced Course					**CS 650 / DIC**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Joachim Weickert
Lecturer	Prof. Dr. Joachim Weickert
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	Related core lecture Computer Vision
Assessment / Exams	• Regular attendance of lecture and tutorial • 50% of all possible points from weekly assignments to be eligible for the final exam are needed • Passing the final exam or the re-exam • The re-exam takes place during the last two weeks before the start of lectures in the following semester
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) 50% theoretical exercises and 50% practical programming assignments
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	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.

Aims / Competences to be developed

Many modern techniques in image processing and computer vision make use of methods based on partial differential equations (PDEs) and variational calculus. Moreover, many classical methods may be reinterpreted as approximations of PDE-based techniques. In this course the students will get an in-depth insight into these methods. For each of these techniques,
they will learn the basic ideas as well as theoretical and algorithmic aspects. Examples from the fields of medical imaging and computer aided quality control will illustrate the various application possibilities.

Content

1. Introduction and Overview
2. Linear Diffusion Filtering
   - 2.1 Basic Concepts
   - 2.2 Numerics
   - 2.3 Limitations and Alternatives
3. Nonlinear Isotropic Diffusion Filtering
   - 3.1 Modeling
   - 3.2 Continuous Theory
   - 3.2 Semidiscete Theory
   - 3.3 Discrete Theory
   - 3.4 Efficient Sequential and Parallel Algorithms
4. Nonlinear Anisotropic Diffusion Filtering
   - 4.1 Modeling
   - 4.2 Continuous Theory
   - 4.3 Discrete Aspects
5. Parameter Selection
6. Variational Methods
   - 6.1 Basic Ideas
   - 6.2 Discrete Aspects
   - 6.3 TV Denoising, Equivalence Results
   - 6.4 Mumford-Shah Segmentation and Diffusion-Reaction Filters
7. Vector- and Matrix-Valued Images
8. Image Sequence Analysis
   - 8.1 Global Methods
   - 8.2 Local Methods
   - 8.3 Combined Local-Global Methods
   - 8.4 Numerical Techniques
9. Continuous-Scale Morphology
   - 9.1 Basic Ideas
   - 9.2 Applications
10. Curvature-Based Morphology
    - 10.1 Basic Ideas
    - 10.2 Applications

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Introduction to Image Acquisition Methods, Advanced Course					**CS 750 / IIAM**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	2	4

Responsible Lecturer	Prof. Dr. Joachim Weickert
Lecturer	N. N.
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate Course / Elective
Entrance requirements	Related core lecture Computer Vision
Assessment / 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 Typ / weekly hours	Lecture 2 h (weekly)
Total workload	120 h = 30 h classes and 90 h private study
Grade of the module	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.

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.

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Correspondence Problems in Computer Vision, Advanced Course					**COPCV**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	At least once every two years	1 Semester	4	6

Responsible Lecturer	Prof. Dr. Joachim Weickert
Lecturer	Prof. Dr. Joachim Weickert
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate Course / Elective
Entrance requirements	Related core lecture Computer Vision, Completed Mathematics for Computer Scientist lectures.
Assessment / Exams	- Regular attendence of lecture and tutorial - Written or oral exam and the end of the course
Course Typ / weekly hours	Lecture 2 h (weekly) Tutorial 2 h (weekly)
Total workload	180 h = 60 h classes and 120 h private study
Grade of the module	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.

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

Additional Information

Teaching language: English

Literature:
Will be announced on the course website

Future Media Internet - FMI					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	WS	1 Semester	4V2Ü	9

Responsible Lecturer	Prof. Dr.-Ing. Thorsten Herfet
Lecturer	Prof. Dr.-Ing. Thorsten Herfet
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Extended Courses
Entrance requirements	For graduate students: none
Assessment / Exams	Weekly exercise sheets, two blocks, each one must be passed individually, oral exam at the end of the modul
Course Typ / weekly hours	Extended Course, 4V2Ü
Total workload	9 CPs = 270 hrs for an average student
Grade of the module	Graded absolute 1.0-n.b. and relative A-F

Aims / Competences to be developed

The course deals with Media Transport over the Internet. After the course students know how data- and mediatransport is solved in today's Internet and have a good understanding of so called erasure channels.
Besides the pure transport protocol design the course complements the fundaments laid in TCI and TCII be introducing state-of-the-art error codes (Van-der-Monde-Codes, Fountain Codes) and by engineering tasks like the design of a Digital PLL.

Content

The course introduces media transmission over packet channels, specifically the Internet. After establishing a Quality of Service framework built on ITU requirements the course models erasure channels without and with memory. Key characteristics like the channel capacity and the minimum redundancy information are derived.

The second part of the course introduces current media transport protocol suites (TCP, UDP, RTP, RTSP) and middleware (ISMA, DLNA, UPnP, DVB-IPI).

In the second half of the course audiovisual coders used in the Internet are introduced (H.264, AAC), state-of-the-art forward error coding schems (Van-der-Monde-Codes, Fountain Codes) are explained and essential elements like a Digital Phase-locked Loop are developed.

Additional Information

Teaching language: English

Literature:
The course will come with a self contained manuscript. The most essential monographs used for and referenced within the manuscript are available in the Computer Science Library of Saarland University.

Automatic Planning, Advanced Course					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1-3		WS	1 Semester	4	9

Responsible Lecturer	Prof. Dr Jörg Hoffmann
Lecturer	Prof. Dr. Jörg Hoffmann
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	Regular attendance of classes and tutorial Paper as well as programming exercises for exam qualification Final exam A re-exam takes place before the start of lectures in the following semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly)
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Aims / Competences to be developed

The students will gain a deep understanding of algorithms used in Automatic Planning for the efficient exploration of large state spaces, from both a theoretical and practical point of view. The programming exercises will familiarize them with the main implementation basis in Automatic Planning. The search algorithms are generic and are relevant also in other CS sub-areas in which large transition systems need to be analyzed.

Content

Automatic Planning is one of the fundamental sub-areas of Artificial Intelligence, concerned with algorithms that can generate strategies of action for arbitrary autonomous agents in arbitrary environments. The course examines the technical core of the current research on solving this kind of problem, consisting of paradigms for automatically generating heuristic functions (lower bound solution cost estimators), as well as optimality-preserving pruning methods. Apart from understanding these techniques themselves, the course explains how to analyze, combine, and compare them.

Starting from an implementation basis provided, students implement their own planning system as part of the course. The course is concluded by a competition between these student systems.

Additional Information

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.

Seminar **Changing Topics**					**CS 500**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	jedes Semester	1 Semester	3	7

Responsible Lecturer	Dean of studies and relevant Professor
Lecturer	Professors oft he Department
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate course / Mandatory Elective
Entrance requirements	Basic knowledge in the field of computer science under focus in the respective seminar.
Assessment / Exams	• Contributions to discussions • Thematic talk • Written elaboration • Final oral examination on the entire scientific area spanned by the seminar
Course Typ / weekly hours	Seminar 3 h (weekly) / groups of up to 20 students
Total workload	210 h = 60 h classes und 150 h private study
Grade of the module	The modalities of the grading will be determined by the responsible professor

Aims / Competences to be developed

At the end of the course students have gained a thorough knowledge of current or foundational aspects of a specific area in computer science.

They attained competences in independently investigating, classifiying, summarizing, discussing, criticizing scientifc issues and presenting scientific findings.

Content
Practical exercising of

- Reflecting on scientific work,
- Analyzing and assessing scientific papers
- Composing scientific abstracts
- Discussing scientific work in a peer group
- Developing common standards for scientific work
- Presentation techniques

Specific focus according to the individual topic of the seminar.

Typical course progression:

- Preparatory meetings to guide selection of individual topics
- Repetitive meetings with discussions of selected contributions
- Talk and elaboration on one of the contributions

Oral exam on entire scientific area spanned by the seminar

Additional Information

Teaching language: English

Literatur:
According to the topic

Master Seminar					**CS 890**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
4	4	jedes Semester	1 Semester	5	12

Responsible Lecturer	Dean of Studies
Lecturer	Professors of the department
Level of the unit / mandatory or not	Master Computer Science Graduate course / Compulsory
Entrance requirements	All mandatory modules except Master seminar and Master thesis
Assessment / Exams	• Presentation of a scientific article of adequate depth in the reading group • Active participation in the discussion in the reading group • Presentation of the planned thesis topic followed by a plenary discussion • Written description of the topic of the Master thesis
Course Typ / weekly hours	Seminar 3h per week (about 15 members)
Total workload	360 h • Hands-on training 2h per week (15 students) • Contact with supervisor 1h per week • Self-study 20h per week
Grade of the module	graded

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".

Additional Information

Teaching language: English

Master Thesis					**CS 899**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
4	4	jedes Semester	1 Semester	2	30

Responsible Lecturer	Professors of the Department
Lecturer	Professors of the Department
Level of the unit / mandatory or not	Master Informatik Graduate / Compulsory
Entrance requirements	Master Seminar
Assessment / 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 Typ / weekly hours	
Total workload	900 h = 50 h contact hours, 850 h private studies
Grade of the module	graded

Aims / Competences to be developed

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

Additional Information

Teaching language: English

Literature:
According to the topic

Tutor					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	3	jedes Semester	1 Semester		4

Responsible Lecturer	Professors of the Department
Lecturer	Professors of the Department
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Elective Compulsory for students being in the foster program
Entrance requirements	Each lecturer selects the tutors for his courses. A prerequisite for becoming a tutor is a very good grade in the relevant course, interest in didactics and an observable talent for didactical work.
Assessment / Exams	The lecturer supervises tutors and gives them feedback regarding their contributions to weekly assignments (creating, finding sample solutions for exisiting eercises), answers to questions on the mailing list as well as correcting the exams. The assistant of the course visits each tutorial once a semester and gives feedback to the tutor as well as to the lecturer. At the end of the semester each students evaluates the work of his/her tutor as a part of the course evaluation.
Course Typ / weekly hours	Tutorial 2 h (weekly) Tutoring groups of up to 20 students

Total workload	A tutor assists a course (usually basic or core lectures) for one semester. This includes the following tasks: 0) Learning the specific didactic aspects of the course matter (4h). 1) Moderating the weekly meetings (90 min each) of a tutorial group 2) Correction of weekly tests, taken in the group 3) Weekly office hours (90 min) for students attending the course. 4) Attending weekly team-meetings with all tutors and lecturers of the course (45 min) 5) Participation in developing sample exercise solutions of the weekly assignments (90 min weekly) 6) Answering incoming questions on the mailing list regarding topics of the course and the weekly assignments (60 min weekly) 7) Getting to grips with the contents of the current lecture (2h weekly) 8) Creating new exercises (1h weekly) 9) Supervising and correcting exams (midterm, final exam, re-exam, 12h each)
Grade of the module	ungraded

Aims / Competences to be developed

Tutors learn how courses are being organized and which methodical aims are being followed. They learn how to communicate complex scientific subject matters to larger groups and in individual meetings.

Before starting their work the students attend one or more colloquia in which they are introduced to the specific didactic aspects of the course matter.

In assisting the course, they learn how to adapt to the different background knowledge and intellectual capicities of the attending students. They get encouraged to communicate complex contexts in a concise and effective way. In addition they get used to communicating subject matters in English.

Content

See above

Additional Information

Teaching Language: Deutsch/Englisch

Soft Skill Seminar					**SSS**
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 -3	3		1 Semester	2	4

Responsible Lecturer	Jennifer Gerling
Lecturer	Jennifer Gerling
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Graduate / Elective
Entrance requirements	
Assessment / Exams	2 hand in presentations - your log - application / cv / ad - scientific text
Course Typ / weekly hours	Blockseminar 120 h 40 h preparation / 40 h course / 40 h private study
Total workload	120 h
Grade of the module	

Aims / Competences to be developed

1. Communication In this part, students learn about the meaning of communication in their every day professional and private life. After an introduction to communication theory including body language and verbal, non-verbal and vocal aspects of communication, there will be exercises dealing with body language, voice sound and team communication, as well as advice concerning communication techniques and handling conflicts.

2. Job Hunting Tailor-made for the students' needs, this is a theoretical and practical training for job application. Students learn about self-assessment, orientation, career planning and the actual application process. The layout and content of a CV and cover letter are discussed, as well as the structure of a job interview, rules concerning conduct and appearance, and advice for assessment centres. Students will enter realistic role play sessions with job interviews and an assessment centre.

3. Scientific Posters Scientists quite often have to present their work as a poster. This part covers the planning phase and the actual realisation, explaining rules for content and layout with respect to the target audience, the use of colours and illustrations, text formatting, as well as special requirements of the print medium as opposed to on-screen presentations. The students are shown examples of existing posters. Finally, they analyse a poster they brought themselves and correct it.

4. Presentation Skills Topics are: how to structure a presentation, designing PowerPoint slides, visual aids and technical equipment, handling questions, timing, dealing with nervousness, how to give proper feedback. Exercises deal with posture, breathing, voice and body language. Students give individual presentations and are video-taped by staff. They get individual feedback and can watch themselves on film. In a second session of presentations, the students can check on their improvement.

5. Time and Self Management Students learn to identify time wasters and to keep an activity log. They are taught how to set work priorities by classifying their goals and arranging them in 4 categories (Eisenhower principle); they learn about action plans and to-do-lists, as well as effective scheduling. Practical exercises introduce creativity techniques (brainstorming, mindmapping®, decision tree) and mnemo techniques.

6. Project Management The following issues are dealt with: the different planning phases, possible problems, communicating in the right way, defining targets, making vague ideas into specific parts of the plan, the right level of detail, network diagrams and Gantt charts, delegating work, guiding a team, risk management, bringing the project to a close, and post-project evaluation. The course also includes a practical exercise.

7. Scientific Writing. This part consists of a detailed lecture, as well as practical exercises and deals with the general structure of a paper and related issues. Students also learn about the process of publishing a paper: rules for submitting a manuscript, dealing with the reviewers' comments etc.

Content

See above

Additional Information

Teaching language:English

Literature:
According to the topic

Language Course					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 4	4		1 Semester	2 or 4	3 or 6

Responsible Lecturer	Dr. Peter Tischer, head of the Language Center
Lecturer	http://www.szsb.uni-saarland.de/mitarbeiter/
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik For each language taught at the center, different levels are offered: beginner, intermediate and advanced level Elective
Entrance requirements	For the beginners level: none French, English, Spanish: assessment test to ascertain the proficiency of each student For all other courses on an advanced level: proof of other language courses or meeting with the lecturer.
Assessment / Exams	Usually exam at the end of the semester and regular attendance (at least 80 % of all classes).
Course Typ / weekly hours	Seminar with 2 - 4 hours of classes each week, independent study with monthly meetings or 4 week intensive courses with 4 h of classes each day. Groups of 6 to 40 students
Total workload	90 h = 30 h classes and 60 h private study 180 h = 60 h classes and 120 h private study
Grade of the module	ungraded

Aims / Competences to be developed

Language skills: grammar, vocabulary, conversation skills.

Content

Depending on course

Additional Information

Teaching language: German and taught language

Literature:
Depending on course

Language Course - German Language Course for Beginners					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 4	4		1 Semester	2 or 4	3 or 6

Responsible Lecturer	NN
Lecturer	NN
Level of the unit / mandatory or not	1.- 4. Semester / international Master students only Elective
Entrance requirements	none
Assessment / Exams	Weekly assignments One presentation Exam at the end of the semester Regular attendance (at least 75% of all classes)
Course Typ / weekly hours	Seminar 6 h of classes each week Groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	ungraded

Aims / Competences to be developed

Students should develop basic skills in

- Reading / understanding German texts
- Understanding spoken German
- Conducting a German conversation
- German Grammar
- Writing German texts

Content

See above

Additional Information

Teaching language: German

Literature:
Depending on course

Language Course - German Language Course / all levels					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 4	4		1 Semester	2 or 4	3 or 6

Responsible Lecturer	NN
Lecturer	NN
Level of the unit / mandatory or not	1.-3. Semester / courses are offered each semester
Entrance requirements	Language test to assesss the proficiency of the student
Assessment / Exams	Weekly assignments One presentation Exam at the end of the semester Regular attendance (at least 75% of all classes)
Course Typ / weekly hours	Seminar 4 h of classes each week Groups of up to 20 students
Total workload	180 h = 60 h of classes and 120 h private study
Grade of the module	ungraded

Aims / Competences to be developed

Students should develop basic skills in

- Reading / understanding German texts
- Understanding spoken German
- Conducting a German conversation
- German Grammar
- Writing German texts

Content

See above

Additional Information

Teaching language:German

Literatur:
Depending on course

Modul Praktikum zum Informationsmanagement					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
		jedes Semester / Beginn jederzeit möglich	1 Semester	4 oder 6	6 oder 9

Responsible Lecturer	Prof. Dr. Schmidt
Lecturer	Prof. Dr. Schmidt und Mitarbeiter
Level of the unit / mandatory or not	Bachelor Informatik Master Informatik Freie Leistungspunkte
Entrance requirements	Programming skills in Java and/or VB.NET
Assessment / Exams	Writing of a report in paper form and passing an oral exam
Course Typ / weekly hours	Internship / 4 SWS oder 6 SWS
Total workload	180 (4 SWS) / 270 (6 SWS) h
Grade of the module	

Aims / Competences to be developed

- improving knowledge on methods and models of Operations Research
- transfer of scientific knowledge into practical solutions
- Independent work on a solution for a given problem (project management)

Content

The internship includes varying tasks from the field of Operations Research and is mainly processed at the chair. Parts of the internship can be done at home by arrangement. The student has her own workstation and is invited to participate in the scientific discussions at the chair. The internship can be started at any time (also in the semester break). The concrete tasks are formulated in cooperation with the supervisor.

Additional Information

Working language: german / english

Literature: Frederick Hillier, Gerald Lieberman, Introduction to Operations Research, edn.10, Mcgraw-Hill, 2014