Module Descriptions

Master Program Computer Science

(11.04.2013)
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.
General rules for primary and master lectures

The following rules apply under normal circumstances. Exceptions are possible and necessary, for example, in the case of the block lectures in March and September.

1) Students are given two opportunities to take the closing examinations: one at the end of the lecture period (end of term) and one at the end of the semester (final). This dual examination option is meant to help students better plan and manage their workload for a semester.

2) Students who take the exam at the end of the lecture period (end of term) may also take the final exam. In this case, the best score of the two exams is recorded.

3) Admission to the end of term / final examinations is normally achieved by collecting a minimum number of points over the semester; they are earned with tests, by turning in homework assignments, or through participation in a mid-term exam, for example.

4) End of term and final exams can be considered exam parts, so that other course achievements, for example those mentioned in point 3 above, can be included in the final grade.

5) The complete and binding examination modalities are posted on a lecture’s website at the beginning of a lecture period.
<table>
<thead>
<tr>
<th>Rs.-Sem.</th>
<th>Modul</th>
<th>CP</th>
<th>SWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>Operating Systems</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Computer Graphics</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Database Systems</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Embedded Systems</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Information Retrieval and Data Mining</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Artificial Intelligence</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Computer Architecture</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Security</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Software Engineering</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Compiler Construction</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Automated Reasoning</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Image Processing and Computer Vision</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Computer Algebra</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Algorithms and Data Structures</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Introduction to Computational Logic</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Geometric Modelling</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Complexity Theory</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Cryptography</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Optimization</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Semantics</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Verification</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Telecommunications I</td>
<td>9</td>
<td>6</td>
</tr>
</tbody>
</table>
Naturwissenschaftlich-Technische Fakultät I
Master-Studiengang Informatik

<table>
<thead>
<tr>
<th>Rs.-Sem.</th>
<th>Modul</th>
<th>CP</th>
<th>SWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>Machine Learning</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Distributed Systems</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Data Networks</td>
<td>9</td>
<td>6</td>
</tr>
</tbody>
</table>

**Advanced Courses (Elective Courses)**

<table>
<thead>
<tr>
<th>Rs.-Sem.</th>
<th>Modul</th>
<th>CP</th>
<th>SWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-3</td>
<td>Computer Architecture 2</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>2-3</td>
<td>Telecommunications II</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>2-3</td>
<td>Automata, Games and Verification</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>2-3</td>
<td>Automated Debugging</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>2-3</td>
<td>Computer Graphics II</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>2-3</td>
<td>Differential Equations in Image Processing and Computer Vision</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>2-3</td>
<td>Introduction to Image Acquisition Methods</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>2-3</td>
<td>Correspondence Problems in Computer Vision</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>2-3</td>
<td>Future Media Internet</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>1-3</td>
<td>Seminar</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Master-Seminar</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Master Thesis</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>1-3</td>
<td>Tutor</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>1-3</td>
<td>Soft Skill</td>
<td>4 o. 6</td>
<td>2 o. 4</td>
</tr>
<tr>
<td>1 - 3</td>
<td>Language Courses</td>
<td>4 o. 6</td>
<td>2 o. 4</td>
</tr>
</tbody>
</table>
Zeile 1: Naturwissenschaftlich-Technische Fakultät I
Zeile 2: Master-Studiengang Informatik

<table>
<thead>
<tr>
<th>Operating Systems, Core Course</th>
<th>CS 551 / OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Studiensem.</td>
<td>5</td>
</tr>
<tr>
<td>Regelstudiensem.</td>
<td>5 - 6</td>
</tr>
<tr>
<td>Turnus</td>
<td>At least once every two years</td>
</tr>
<tr>
<td>Dauer</td>
<td>1 Semester</td>
</tr>
<tr>
<td>SWS</td>
<td>6</td>
</tr>
<tr>
<td>ECTS-Punkte</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Peter Druschel, Ph.D.
Dozent/inn/en: Prof. Peter Druschel, Ph.D.
Björn Brandenburg, Ph.D.

Zuordnung zum Curriculum: Graduate course / Mandatory Elective

Zulassungsvoraussetzungen
Leistungskontrollen / Prüfungen: 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.

Lehrveranstaltungen / SWS: Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Arbeitsaufwand: 270 h = 90 h of classes and 180 h private study


Lernziele / Kompetenzen
Introduction to the principles, design, and implementation of operating systems
Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Naturwissenschaftlich-Technische Fakultät I
Master-Studiengang Informatik

Computer Graphics, Core Course | CS 552 / CG
---|---
Studiensem. | 5
Regelstudiensem. | 5 - 6
Turnus | At least once every two years
Dauer | 1 Semester
SWS | 6
ECTS-Punkte | 9

Modulverantwortliche/r | Prof. Dr. Philipp Slusallek
Dozent/inn/en | Prof. Dr. Philipp Slusallek, Prof. Dr. Hans-Peter Seidel

Zuordnung zum Curriculum
- Bachelor Informatik
- Master Informatik
- Graduate course / Mandatory Elective

Zulassungsvoraussetzungen
- For graduate students: none

Leistungskontrollen / Prüfungen
- Successful completion of at least 50% of the exercises
- Successful 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.

Lehrveranstaltungen / SWS
- Lecture 4 h (weekly)
- Tutorial 2 h (weekly)
- Tutorials in groups of up to 20 students

Arbeitsaufwand | 270 h = 90 h of classes and 180 h private study

Modulnote
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen
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.
Inhalt

- Fundamentals of digital image synthesis
  - Physical 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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Database Systems, Core Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r
Prof. Dr. Jens Dittrich

Dozent/inn/en
Prof. Dr. Jens Dittrich

Zuordnung zum Curriculum
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen
especially Information Systems
For graduate students:
• motivation for databases and database management systems;
• the relational data model;
• relational query languages, particularly relational algebra and SQL; XML;
• solid programming skills in Java

Leistungskontrollen / Prüfungen
• Passing a two-hour written exam at the end of the semester
• Successful demonstration of programming project (teams of 2 students are allowed)

Grades are based on written exam (100 points); successful demonstration of the programming project is a requirement for the admission to the exam. It is possible to obtain up to ca. 20 bonus points for the programming project (for efficient implementations and the implementation of advanced query optimization techniques)

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

Lehrveranstaltungen / SWS
Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Arbeitsaufwand
270 h = 90 h of classes and 180 h private study

Modulnote
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.
Lernziele / Kompetenzen

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, a DBMS kernel will be implemented and its performance evaluated. The goal of this implementation project 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. Moreover, an important goal of this project - and the course as a whole - is to communicate the essential difference between being a mere programmer and being a systems expert: The techniques taught in the course should allow the participant, starting the implementation project with a naive prototype, to attain query processing performance improvements of many orders of magnitude, far beyond what could be achieved by good programming alone.

Inhalt

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

- storage media (disk, flash, main memory, caches)
- 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, LSM and stepped merge trees, read- and write-optimized indexing, data warehouse indexing, text indexing, main-memory indexes, sparse and dense, direct and indirect, clustered and unclustered, main memory versus disk and/or flash-based)
- processing models (operator, push and pull, block-based, vectorized, compiled)
- processing implementations (join algorithms for relational, spatial, and multidimensional data, grouping and early aggregation, filtering)
- query processing (scanning, plan computation, SIMD)
- query optimization (query rewrite, cost models, cost-based optimization, join order, plan enumeration)
- data recovery (single versus multiple instance, logging)
- parallelization of data and queries (horizontal and vertical partitioning, shared-nothing, replication, distributed query processing, NoSQL, MapReduce and Hadoop)
- read-optimized system concepts (search engines, data warehouses, OLAP, ad-hoc analytics)
- write-optimized system concepts (OLTP, streaming data, moving objects)
- management of geographical data (GIS, google maps)

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Naturwissenschaftlich-Technische Fakultät I
Master-Studiengang Informatik

<table>
<thead>
<tr>
<th>Embedded Systems, Core Course</th>
<th>CS 650 /ES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Studiensem.</td>
<td>Regelstudiensem.</td>
</tr>
<tr>
<td>5</td>
<td>5 - 6</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Bernd Finkbeiner, Ph.D
Dozent/inn/en: Prof. Dr. Reinhard Wilhelm, Prof. Bernd Finkbeiner, Ph.D

Zuordnung zum Curriculum:
- Bachelor Informatik
- Master Informatik
- Graduate course / Mandatory Elective

Zulassungsvoraussetzungen

Leistungskontrollen / Prüfungen:
- 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.

Lehrveranstaltungen / SWS:
- 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.

Arbeitsaufwand: 270 h = 90 h classes and 180 h private study

Modulnote:
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen:
The students should learn methods for the design, the implementation, and the validation of safety-critical embedded systems.
Inhalt

Embedded Computer Systems are components of a technical system, e.g. an airplane, 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.

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Naturwissenschaftlich-Technische Fakultät I
Master-Studiengang Informatik

<table>
<thead>
<tr>
<th>Information Retrieval and Data Mining, Core Course</th>
<th>CS 555 / IRDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Studiensem.</td>
<td>Regelstudiensem.</td>
</tr>
<tr>
<td>5</td>
<td>5 - 6</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r
Prof. Dr. Gerhard Weikum

Dozent/inn/en
Prof. Dr. Gerhard Weikum

Zuordnung zum Curriculum
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen
For graduate students: none

Leistungskontrollen / Prüfungen
- Regular attendance of classes and tutorials
- Passing 2 of 3 written exams (midterm, final and re-exam)
- Presentation of a solution during a tutorial (at least once)
- For each additional presentation up to 3 bonus points can be gained
- Passing the practical exercises (teams of up to two students)
- Up to 3 bonus points can be gained for the overall quality of the solutions
- The re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS
Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Arbeitsaufwand
270 h = 90 h of classes and 180 h private study

Modulnote
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen
The lecture teaches mathematical models and algorithms that form the basis for search engines for the Web, intranets, and digital libraries and for data mining and analysis tools.

Inhalt
Information Retrieval and Data Mining are technologies for searching, analyzing and automatically
organizing text documents, multi-media documents, and structured or semistructured data. The course teaches mathematical models and algorithms that form the basis for search engines for the Web, intranets, and digital libraries and for data mining and analysis tools. The fundamentals are models and methods from linear algebra and regression (e.g. singular-value decomposition) as well as probability theory and statistics (e.g. Bayesian networks and Markov chains). The exercises include practical tasks for the implementation of a simple search engine in Java.

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Artificial Intelligence, Core Course | CS 556 / AI
---|---
Studiensem. | Regelstudiensem. | Turnus | Dauer | SWS | ECTS-Punkte
5 | 5 - 6 | At least once every two years | 1 Semester | 6 | 9

Modulverantwortliche/r
Prof. Dr. Wolfgang Wahlster

Dozent/inn/en
Prof. Dr. Wolfgang Wahlster
Prof. Dr. Jörg Hoffmann

Zuordnung zum Curriculum
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen
For graduate students: none

Leistungskontrollen / Prüfungen
- 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.

Lehrveranstaltungen / SWS
Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Arbeitsaufwand
270 h = 90 h of classes and 180 h private study

Modulnote
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauem Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen
Knowledge about the fundamentals of artificial intelligence

Inhalt
Problem-solving:
- Uninformed- and informed search procedures
- Adversarial search
- Knowledge and reasoning:
- First-order logic, Inference in first-order logic
- Knowledge representation

Planning:
- Planning
- Planning and acting in the real world
Uncertain knowledge and reasoning:
- Uncertainty
- Probabilistic reasoning
- Simple & complex decisions

Learning:
- Learning from observations
- Knowledge in learning
- Statistical learning methods
- Reinforcement learning

Communicating, perceiving, and acting:
- Communication
- Natural language processing
- Perception

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Naturwissenschaftlich-Technische Fakultät I  
Master-Studiengang Informatik

<table>
<thead>
<tr>
<th>Computer Architecture, Core Course</th>
<th>CS 558 / CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Studiensem.</td>
<td>Regelstudiensem.</td>
</tr>
<tr>
<td>5</td>
<td>5 - 6</td>
</tr>
</tbody>
</table>

**Modulverantwortliche/r**  
Prof. Dr. W.-J. Paul

**Dozent/inn/en**  
Prof. Dr. W.-J. Paul

**Zuordnung zum Curriculum**  
Bachelor Informatik  
Master Informatik  
Graduate course / Mandatory Elective

**Zulassungsvoraussetzungen**  
For graduate students: none

**Leistungskontrollen / Prüfungen**  
**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.

**Lehrveranstaltungen / SWS**  
Lecture 4 h (weekly)  
Tutorial 2 h (weekly)  
Tutorials in groups of up to 20 students

**Arbeitsaufwand**  
270 h = 90 h of classes and 180 h private study

**Modulnote**  
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

**Lernziele / Kompetenzen**

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.

**Inhalt**

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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
## Security, Core Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

### Modulverantwortliche/r
Prof. Dr. Michael Backes

### Dozent/inn/en
Prof. Dr. Michael Backes

### Zuordnung zum Curriculum
- Bachelor Informatik
- Master Informatik
- Graduate course / Mandatory Elective

### Zulassungsvoraussetzungen
For graduate students: none

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

### Lehrveranstaltungen / SWS
- Lecture 4 h (weekly)
- Tutorial 2 h (weekly)
- Tutorials in groups of up to 20 students

### Arbeitsaufwand
270 h = 90 h of classes and 180 h private study

### Modulnote
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekanntgegeben.

### Lernziele / Kompetenzen
Description, assessment, development and application of security mechanisms, techniques and tools.

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

### Weitere Informationen
Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Software Engineering, Core Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Dr. Andreas Zeller
Dozent/inn/en: Prof. Dr. Andreas Zeller

Zuordnung zum Curriculum: Graduate course / Mandatory Elective

Zulassungsvoraussetzungen: For graduate students: none

Leistungskontrollen / Prüfungen:
- Successful project completion (including deliverables such as requirements, design, implementation)
- Successful project demonstration
- Regular attendance of classes
- Passing the final exam

Lehrveranstaltungen / SWS:
- Lecture 2 h (weekly)
- Project 4 h (weekly)
- Project work in teams of 4–7 students

Arbeitsaufwand: 270 h = 90 h of classes and 180 h private study


Lernziele / Kompetenzen:
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)
Naturwissenschaftlich-Technische Fakultät I
Master-Studiengang Informatik

- 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 management)

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Compiler Construction, Core Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r
Prof. Dr. Reinhard Wilhelm, Prof. Dr. Sebastian Hack

Dozent/inn/en
Prof. Dr. Reinhard Wilhelm, Prof. Dr. Sebastian Hack

Zuordnung zum Curriculum
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen
For graduate students: none

Leistungskontrollen / Prüfungen
- 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.

Lehrveranstaltungen / SWS
Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Arbeitsaufwand
270 h = 90 h of classes and 180 h private study

Modulnote
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen
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.

Inhalt
Lexical, syntactic, semantic analysis of source programs, code generation for abstract and real machines, efficiency-improving program transformations. Generative methods for compilation subtasks.

Weitere Informationen
Unterrichtssprache: Englisch

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

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Dr. Christoph Weidenbach
Dozent/inn/en: Prof. Dr. Christoph Weidenbach

Zuordnung zum Curriculum: Graduate course / Mandatory Elective

Zulassungsvoraussetzungen: CS 575 ICL

Leistungskontrollen / Prüfungen:
- 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.

Lehrveranstaltungen / SWS:
- Lecture 4 h (weekly)
- Tutorial 2 h (weekly)
- Tutorials in groups of up to 20 students

Arbeitsaufwand: 270 h = 90 h of classes and 180 h private study


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

Inhalt:
- 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
Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Image Processing and Computer Vision, Core Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r
Prof. Dr. Joachim Weickert

Dozent/inn/en
Prof. Dr. Joachim Weickert

Zuordnung zum Curriculum
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen
For graduate students: none

Leistungskontrollen / Prüfungen
- Regular attendance of classes and tutorials.
- At least 50% of all possible points from the weekly assignments have to be gained to qualify for the 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.

Lehrveranstaltungen / SWS
Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Arbeitsaufwand
270 h = 90 h of classes and 180 h private study

Modulnote
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen
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.
Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Naturwissenschaftlich-Technische Fakultät I  
Master-Studiengang Informatik

<table>
<thead>
<tr>
<th>Computer Algebra, Core Course</th>
<th>CS 573 / CA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Studiensem.</td>
<td>5</td>
</tr>
<tr>
<td>Regelstudiensem.</td>
<td>5 - 6</td>
</tr>
<tr>
<td>Turnus</td>
<td>At least once every two years</td>
</tr>
<tr>
<td>Dauer</td>
<td>1 Semester</td>
</tr>
<tr>
<td>SWS</td>
<td>6</td>
</tr>
<tr>
<td>ECTS-Punkte</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Dr. Frank-Olaf Schreyer  
Dozent/inn/en: Prof. Dr. Frank-Olaf Schreyer

Zuordnung zum Curriculum:  
Bachelor Informatik  
Master Informatik  
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen:  
For graduate students: none

Leistungskontrollen / Prüfungen:  
• Regular attendance of classes and tutorials  
• Solving the exercises, passing the midterm and the final exam.  
• Grade: 20% exercises, 30% midterm, 50% final exam.  
• A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS:  
Lecture 4 h (weekly)  
Tutorial 2 h (weekly)  
Tutorials in groups of up to 20 students

Arbeitsaufwand:  
270 h = 90 h of classes and 180 h private study

Modulnote:  
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben

Lernziele / Kompetenzen:  
Solving problems occurring in computer algebra practice  
The theory behind algorithms
Inhalt

Arithmetic and algebraic systems of equations in geometry, engineering and natural sciences
- integer and modular arithmetics, prime number tests
- polynomial 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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Algorithms and Data Structures, Core Course | CS 574 / A&D

<table>
<thead>
<tr>
<th>Study Semester</th>
<th>Regular Study Semester</th>
<th>Turnus</th>
<th>Duration</th>
<th>SWS</th>
<th>ECTS Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

**Module Responsible**
Prof. Dr. Kurt Mehlhorn

**Lecturers**
Prof. Dr. Kurt Mehlhorn, Prof. Dr. Raimund Seidel

**Affiliation to Curriculum**
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

**Entry Requirements**
For graduate students: C, C++, Java

**Assessment/Examinations**
- 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.

**Teaching Activities/SWS**
- Lecture 4 h (weekly)
- Tutorial 2 h (weekly)
- Tutorials in groups of up to 20 students

**Workload**
270 h = 90 h of classes and 180 h private study

**Module Grade**
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben

**Learning Objectives/Competencies**
The students know standard algorithms for typical problems in the areas graphs, computational geometry, strings and optimization. Furthermore they master a number of methods and data-structures to develop efficient algorithms and analyze their running times.
Inhalt

- 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.)

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
### Introduction to Computational Logic, Core Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

### Modulverantwortliche/r
- Prof. Dr. Gert Smolka

### Dozent/inn/en
- Prof. Dr. Gert Smolka

### Zuordnung zum Curriculum
- Bachelor Informatik
- Master Informatik
- Graduate course / Mandatory Elective

### Zulassungsvoraussetzungen

### Leistungskontrollen / Prüfungen
- Regular attendance of classes and tutorials.
- Passing the midterm and the final exam.

### Lehrveranstaltungen / SWS
- Lecture 4 h (weekly)
- Tutorial 2 h (weekly)
- Tutorials in groups of up to 20 students

### Arbeitsaufwand
- 270 h = 90 h of classes and 180 h private study

### Modulnote

### Lernziele / Kompetenzen
- 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
Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
<table>
<thead>
<tr>
<th>Geometric Modeling, Core Course</th>
<th>CS 576 / GM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Studiensem.</td>
<td>Regelstudiensem.</td>
</tr>
<tr>
<td>5</td>
<td>5 - 6</td>
</tr>
</tbody>
</table>

**Modulverantwortliche/r**
Prof. Dr. Hans-Peter Seidel

**Dozent/inn/en**
Prof. Dr. Hans-Peter Seidel,
Prof. Dr. Philipp Slusallek
Dr. Michael Wand
Dr. Tino Weinkauf
Dr. Klaus Hildebrandt

**Zuordnung zum Curriculum**
Graduate course / Mandatory Elective

**Zulassungsvoraussetzungen**
For graduate students: none

**Leistungskontrollen / Prüfungen**
- Weekly Assignments (10% bonus for final exam; bonus points can only improve the grade; they do not affect passing)
- Final exam (100%, but bonus from assignments will be added)
- A re-exam takes place at the end of the semester break or early in the next semester.
- The students can take either the final exam or the re-exam, or both of them. The better grade will become the final grade (this includes passing at all), a worse outcome in one of two exams taken will not be taken into account in any way.

**Lehrveranstaltungen / SWS**
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.

**Arbeitsaufwand**
270 h = 90 h of classes and 180 h private study

**Modulnote**
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.
Lernziele / Kompetenzen

Learning working knowledge of theoretical and practical methods for solving geometric modeling problems on a computer. From a broader perspective: Learning how to represent 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).

Inhalt

- Differential geometry Fundamentals
- Interpolation and Approximation
- Polynomial Curves
- Bezier and Rational Bezier Curves
- B-splines, NURBS
- Spline Surfaces
- Subdivision and Multiresolution Modeling
- Variational, Implicit, Point-Based Models
- Shape Analysis and Geometry Processing

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Complexity Theory, Core Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Dr. Markus Bläser
Dozent/inn/en: Prof. Dr. Markus Bläser, Prof. Dr. Raimund Seidel

Zuordnung zum Curriculum
- Bachelor Informatik
- Master Informatik
- Graduate course / Mandatory Elective

Zulassungsvoraussetzungen
For graduate students: none

Leistungskontrollen / Prüfungen
- Regular attendance of classes and tutorials
- The grades are built from the grades in the midterm exam (40%) and the exam after the end of semester (60%)
- To be admitted to an exam, one has to get at least half of the points that can be obtained so far in the home works.
- A re-exam takes place during the last two weeks before the start of lectures in the following semester.

Lehrveranstaltungen / SWS
- Lecture 4 h (weekly)
- Tutorial 2 h (weekly)
- Tutorials in groups of up to 20 students

Arbeitsaufwand
270 h = 90 h of classes and 180 h private study

Modulnote
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben

Lernziele / Kompetenzen
The aim of this lecture is to understand important concepts and methods of computational complexity theory.
The second part of the lecture teaches the fundamentals to understand recent topics and results in computational complexity theory.
Inhalt

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-Szelepcsényi theorem, polynomial time hierarchy, relativization parity and the polynomial methods, Valiant-Vazirani theorem, counting problems and classes, probabilistic computations, graph isomorphism and interactive proofs. If time permits, more advanced topics like derandomization probabilistic checkable proofs, or optimisation problems and approximation will be covered.

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Arora, Barak: Computational Complexity, Cambridge University Press
### Cryptography, Core Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

**Modulverantwortliche/r**
Prof. Dr. Michael Backes

**Dozent/inn/en**
Prof. Dr. Michael Backes

**Zuordnung zum Curriculum**
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

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

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

**Lehrveranstaltungen / SWS**
- Lecture 4 h (weekly)
- Tutorial 2 h (weekly)
- Tutorials in groups of up to 20 students

**Arbeitsaufwand**
270 h = 90 h of classes and 180 h private study

**Modulnote**
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

**Lernziele / Kompetenzen**
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.

**Inhalt**
- 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
Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Optimization, Core Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

**Modulverantwortliche/r**
Prof. Dr. Kurt Mehlhorn

**Dozent/inn/en**
Prof. Dr. Kurt Mehlhorn, Dr. Khaled Elbassioni

**Zuordnung zum Curriculum**
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

**Zulassungsvoraussetzungen**
For graduate students: none

**Leistungskontrollen / Prüfungen**
- Regular attendance of classes and tutorials
- Solving accompanying exercises, successful participation 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.

**Lehrveranstaltungen / SWS**
Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

**Arbeitsaufwand**
270 h = 90 h of classes and 180 h private study

**Modulinote**
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben

**Lernziele / Kompetenzen**
The students learn to model and solve optimization problems from theory as from the real world
Inhalt

- 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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Semantics, Core Course | CS 580 / SEM
--- | ---
Studiensem. | 5
Regelstudiensem. | 5 - 6
Turnus | At least once every two years
Dauer | 1 Semester
SWS | 6
ECTS-Punkte | 9

**Modulverantwortliche/r** | Prof. Dr. Gert Smolka
**Dozent/inn/en** | Prof. Dr. Gert Smolka

**Zuordnung zum Curriculum**
- Bachelor Informatik
- Master Informatik
- Graduate course / Mandatory Elective

**Zulassungsvoraussetzungen**
For graduate students: core lecture Introduction to Computational Logic

**Leistungskontrollen / Prüfungen**
- Regular attendance of classes and tutorials.
- Passing the midterm and the final exam

**Lehrveranstaltungen / SWS**
- Lecture 4 h (weekly)
- Tutorial 2 h (weekly)
- Tutorials in groups of up to 20 students

**Arbeitsaufwand**
270 h = 90 h of classes and 180 h private study

**Modulnote**
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben

**Lernziele / Kompetenzen**
Understanding of
- Logical structure of programming languages
- Formal models of programming languages
- Type and module systems for programming languages

**Inhalt**
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
Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Verification, Core Course

<table>
<thead>
<tr>
<th>Study Semester</th>
<th>Regular Study Semester</th>
<th>Turnus</th>
<th>Duration</th>
<th>SWS</th>
<th>ECTS Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Module Responsible Person

Prof. Dr. Holger Hermanns

Instructors

Prof. Dr. Holger Hermanns, Prof. Bernd Finkbeiner, Ph.D

Curriculum Assignment

Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Admission Requirements

For graduate students: none

Examination Requirements / Examinations

- 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.

Teaching Activities / SWS

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

Working Hours

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

Module Grade

Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Learning Objectives / Competencies

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.
Inhalt

- 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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Telecommunications I, Core Course | TC I

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Dr.-Ing. Thorsten Herfet

Dozent/inn/en: Prof. Dr.-Ing. Thorsten Herfet

Zuordnung zum Curriculum:
- Bachelor Informatik
- Master Informatik
- Graduate course / Mandatory Elective

Zulassungsvoraussetzungen: 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.

Leistungskontrollen / Prüfungen:
- 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.

Lehrveranstaltungen / SWS:
- Lecture 4 h (weekly)
- Tutorial 2 h (weekly)
- Tutorials in groups of up to 20 students

Arbeitsaufwand: 270 h = 90 h of classes and 180 h private study

Modulnote: Final exam mark

Lernziele / Kompetenzen:
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.
Inhalt

As the basic principle, the course will give an introduction into the various building blocks that modern telecommunication systems do incorporate. Sources, sinks, source and channel coding, modulation and multiplexing are the major keywords but we will also deal with dedicated pieces like A/D- and D/A-converters and quantizers in a little bit more depth. The course will refresh the basic transformations (Fourier, Laplace) that give access to system analysis in the frequency domain, it will introduce derived transformations (z, Hilbert) for the analysis of discrete systems and modulation schemes and it will briefly introduce algebra on finite fields to systematically deal with error correction schemes that play an important role in modern communication systems.

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Machine Learning, Core Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Dr. Matthias Hein
Dozent/inn/en: Prof. Dr. Matthias Hein

Zuordnung zum Curriculum: Bachelor Informatik, Master Informatik, Graduate course / Mandatory Elective

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

Leistungskontrollen / Prüfungen:
- Regular attendance of classes and tutorials.
- 50% of all points of the exercises which are so far possible have to be obtained in order to qualify for the exam.
- Passing 1 out of 2 exams (final, re-exam).

Lehrveranstaltungen / SWS:
- Lecture 4 h (weekly)
- Tutorial 2 h (weekly)
- Tutorials in groups of up to 20 students

Arbeitsaufwand: 270 h = 90 h of classes and 180 h private study


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

Inhalt:
- 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
Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Distributed Systems, Core Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Peter Druschel, Ph.D.

Dozent/inn/en: Prof. Peter Druschel, Ph.D.
Allen Clement, Ph.D.

Zuordnung zum Curriculum: Graduate course / Mandatory Elective

Zulassungsvoraussetzungen: Operating systems or concurrent programming.

Leistungskontrollen / Prüfungen:
- 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.

Lehrveranstaltungen / SWS:
- Lecture 4 h (weekly)
- Tutorial 2 h (weekly)

Arbeitsaufwand: 270 h = 90 h of classes and 180 h private study


Lernziele / Kompetenzen:
Introduction to the principles, design, and implementation of distributed systems

Inhalt:
- 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.
Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Data Networks, Core Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

**Modulverantwortliche/r**
Prof. Dr. Holger Hermanns

**Dozent/inn/en**
Prof. Dr. Holger Hermanns

**Zuordnung zum Curriculum**
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

**Zulassungsvoraussetzungen**
For graduate students: none

**Leistungskontrollen / Prüfungen**
- 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.

**Lehrveranstaltungen / SWS**
Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

**Arbeitsaufwand**
270 h = 90 h of classes and 180 h private study

**Modulnote**
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

**Lernziele / Kompetenzen**
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.
Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
### Computer Architecture 2, Advanced Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

**Modulverantwortliche/r**

Prof. Dr. W. J. Paul

**Dozent/inn/en**

Prof. Dr. W. J. Paul

**Zuordnung zum Curriculum**

Bachelor Informatik  
Master Informatik  
Graduate / Mandatory Elective

**Zulassungsvoraussetzungen**

Related core lecture Computer Architecture

**Leistungskontrollen / Prüfungen**

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

**Lehrveranstaltungen / SWS**

Lecture 4 h weekly, 50-100 students  
Tutorials 2 h weekly, up to 20 students

**Arbeitsaufwand**

270 hours = 90 h classes and 180 h private study

**Modulnote**

Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

**Lernziele / Kompetenzen**

After this lecture students know how to design IEEE compatible floating point units and some form of parallel computer system.
Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Telecommunications II, Advanced Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r
Prof. Dr.-Ing. Thorsten Herfet

Dozent/inn/en
Prof. Dr.-Ing. Thorsten Herfet

Zuordnung zum Curriculum
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen
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

Leistungskontrollen / Prüfungen
Regular attendance of classes and tutorials Passing the final exam
Oral exam directly succeeding the course. Eligibility:
Weekly excercises / 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.

Lehrveranstaltungen / SWS
Lecture 4 h (weekly)
Tutorial 2 h (weekly)
Tutorials in groups of up to 20 students

Arbeitsaufwand
270 hours = 90 h classes and 180 h private study

Modulnote
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.
Lernziele / Kompetenzen

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.

Inhalt

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 802.11 (with the TGe Quality of Service extensions), 802.16a and the terrestrial DVB system (DVB-T, DVB-H).

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Automata, Games and Verification, Advanced Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Bernd Finkbeiner, PhD
Dozent/inn/en: Prof. Bernd Finkbeiner, PhD

Zuordnung zum Curriculum:
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen:

Leistungskontrollen / Prüfungen:
- 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.

Lehrveranstaltungen / SWS:
Lecture 2 h (weekly)
Tutorial 2 h (weekly)

Arbeitsaufwand:
180 h = 60 h classes and 120 h private study


Lernziele / Kompetenzen:
The students will gain a deep understanding of the automata-theoretic background of automated verification and program synthesis.
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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Automated Debugging, Advanced Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Dr. Andreas Zeller
Dozent/inn/en: Prof. Dr. Andreas Zeller

Zuordnung zum Curriculum:
- Bachelor Informatik
- Master Informatik
- Graduate / Mandatory Elective

Zulassungsvoraussetzungen:
Programming skills as acquired at the Bachelor level

Leistungskontrollen / Prüfungen:
- 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.

Lehrveranstaltungen / SWS:
- Lecture 2 h (weekly)
- Tutorial 2 h (weekly)

Arbeitsaufwand:
180 h = 60 h classes and 120 h private study

Modulnote:
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen:
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?
Inhalt

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?

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Computer Graphics II, Advanced Course  
Realistic Image Synthesis

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Dr. Philipp Slusallek
Dozent/inn/en: Prof. Dr. Philipp Slusallek

Zuordnung zum Curriculum:
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective

Zulassungsvoraussetzungen:
Related core lecture Computer Graphics

Leistungskontrollen / Prüfungen:
- 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.

Lehrveranstaltungen / SWS:
Lecture 4 h (weekly)
Tutorial 2 h (weekly)

Arbeitsaufwand:
270 h = 90 h classes and 180 h private study

Modulnote

Lernziele / Kompetenzen

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.
Inhalt
- 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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet
Differential Equations in Image Processing and Computer Vision, Advanced Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r                      Prof. Dr. Joachim Weickert
Dozent/inn/en                                Prof. Dr. Joachim Weickert
Zuordnung zum Curriculum
Bachelor Informatik
Master Informatik
Graduate course / Mandatory Elective
Zulassungsvoraussetzungen Related core lecture Computer Vision
Leistungskontrollen / Prüfungen
• 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
Lehrveranstaltungen / SWS
Lecture 4 h (weekly)
Tutorial 2 h (weekly)
50% theoretical exercises and 50% practical programming assignments
Arbeitsaufwand 270 h = 90 h of classes and 180 h private study

Lernziele / Kompetenzen

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.
Inhalt
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
   3.3 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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Introduction to Image Acquisition Methods, Advanced Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r

Prof. Dr. Joachim Weickert

Dozent/inn/en

N. N.

Zuordnung zum Curriculum

Bachelor Informatik
Master Informatik
Graduate Course / Elective

Zulassungsvoraussetzungen

Related core lecture Computer Vision

Leistungskontrollen / Prüfungen

- 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.

Lehrveranstaltungen / SWS

Lecture 2 h (weekly)

Arbeitsaufwand

120 h = 30 h classes and 90 h private study

Modulnote

Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

Lernziele / Kompetenzen

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.
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.

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
## Correspondence Problems in Computer Vision, Advanced Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5 - 6</td>
<td>At least once every two years</td>
<td>1 Semester</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

**Modulverantwortliche/r**  
Prof. Dr. Joachim Weickert

**Dozent/inn/en**  
Prof. Dr. Joachim Weickert

**Zuordnung zum Curriculum**  
Bachelor Informatik  
Master Informatik  
Graduate Course / Elective

**Zulassungsvoraussetzungen**  
Related core lecture Computer Vision,  
Completed Mathematics for Computer Scientist lectures.

**Leistungskontrollen / Prüfungen**  
- Regular attendance of lecture and tutorial  
- Written or oral exam and the end of the course

**Lehrveranstaltungen / SWS**  
Lecture 2 h (weekly)  
Tutorial 2 h (weekly)

**Arbeitsaufwand**  
180 h = 60 h classes and 120 h private study

**Modulnote**  
Wird aus Leistungen in Klausuren, Übungen und praktischen Aufgaben ermittelt. Die genauen Modalitäten werden vom Modulverantwortlichen bekannt gegeben.

## Lernziele / Kompetenzen

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 correspondence 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.
Naturwissenschaftlich-Technische Fakultät I
Master-Studiengang Informatik

Inhalt
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 Linear 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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet.
Future Media Internet - FMI

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 3</td>
<td>3</td>
<td>WS</td>
<td>1 Semester</td>
<td>4V2Ü</td>
<td>9</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Prof. Dr.-Ing. Thorsten Herfet
Dozent/inn/en: Prof. Dr.-Ing. Thorsten Herfet

Zuordnung zum Curriculum:
- Bachelor Informatik
- Master Informatik
- Extended Courses

Zulassungsvoraussetzungen: For graduate students: none
Leistungskontrollen / Prüfungen:
- Weekly exercise sheets, two blocks, each one must be passed individually, oral exam at the end of the module

Lehrveranstaltungen / SWS: Extended Course, 4V2Ü
Arbeitsaufwand: 9 CPs = 270 hrs for an average student

Modulnote: Graded absolute 1.0-n.b. and relative A-F

Lernziele / Kompetenzen

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 by introducing state-of-the-art error codes (Van-der-Monde-Codes, Fountain Codes) and by engineering tasks like the design of a Digital PLL.

Inhalt

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 schemas (Van-der-Monde-Codes, Fountain Codes) are explained and essential elements like a Digital Phase-locked Loop are developed.
Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:
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.
Seminar Changing Topics | CS 500
--- | ---
Studiensem. | Regelstudiensem. | Turnus | Dauer | SWS | ECTS-Punkte
5 | 5 - 6 | jedes Semester | 1 Semester | 3 | 7

Modulverantwortliche/r | Dean of studies and relevant Professor
Dozent/inn/en | Professors of the Department
Zuordnung zum Curriculum | Bachelor Informatik
| Master Informatik
| Graduate course / Mandatory Elective
Zulassungsvoraussetzungen | Basic knowledge in the field of computer science under focus in the respective seminar.
Leistungskontrollen / Prüfungen
- Contributions to discussions
- Thematic talk
- Written elaboration
- Final oral examination on the entire scientific area spanned by the seminar
Lehrveranstaltungen / SWS | Seminar 3 h (weekly) / groups of up to 20 students
Arbeitsaufwand | 210 h = 45 h classes und 165 h private study
Modulnote | Die Modalitäten der Notenvergabe werden vom verantwortlichen Hochschullehrer festgelegt.

Lernziele / Kompetenzen
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, classifying, summarizing, discussing, criticizing scientific issues and presenting scientific findings.
Inhalt
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

Weitere Informationen

Unterrichtssprache: Englisch

Literatur:
According to the topic
## Master Thesis CS 899

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>1-3</td>
<td>jedes Semester</td>
<td>1 Semester</td>
<td>2</td>
<td>30</td>
</tr>
</tbody>
</table>

### Modulverantwortliche/r
Professors of the Department

### Dozent/inn/en
Professors of the Department

### Zuordnung zum Curriculum
Master Informatik
Graduate / Compulsory

### Zulassungsvoraussetzungen
Master Seminar

### Leistungskontrollen / Prüfungen
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.

### Lehrveranstaltungen / SWS
Arbeitsaufwand: 900 h = 50 h contact hours, 850 h private studies

### Modulnote
graded

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

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

### Weitere Informationen
Unterrichtssprache: Englisch

Literatur:
According to the topic
Tutor

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>jedes Semester</td>
<td>1 Semester</td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

Modulverantwortliche/r
Professors of the Department

Dozent/inn/en
Professors of the Department

Zuordnung zum Curriculum
Bachelor Informatik
Master Informatik
Elective
Compulsory for students being in the foster program

Zulassungsvoraussetzungen
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.

Leistungskontrollen / Prüfungen
The lecturer supervises tutors and gives them feedback regarding their contributions to weekly assignments (creating, finding sample solutions for existing exercises), 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 student evaluates the work of his/her tutor as a part of the course evaluation.

Lehrveranstaltungen / SWS
Tutorial 2 h (weekly)
Tutoring groups of up to 20 students
Arbeitsaufwand

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).

Modulnote

ungraded

Lernziele / Kompetenzen

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 capacities 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.

Inhalt

See above
Weitere Informationen

Unterrichtssprache: Deutsch/Englisch
Naturwissenschaftlich-Technische Fakultät I
Master-Studiengang Informatik

<table>
<thead>
<tr>
<th>Soft Skill Seminar</th>
<th>SSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Studiensem.</td>
<td>1-3</td>
</tr>
<tr>
<td>Regelstudiensem.</td>
<td>1-3</td>
</tr>
<tr>
<td>Turnus</td>
<td>1 Semester</td>
</tr>
<tr>
<td>Dauer</td>
<td>2</td>
</tr>
<tr>
<td>SWS</td>
<td>4</td>
</tr>
<tr>
<td>ECTS-Punkte</td>
<td></td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: Jennifer Gerling
Dozent/inn/en: Jennifer Gerling

Zuordnung zum Curriculum
Bachelor Informatik
Master Informatik
Graduate / Elective

Zulassungsvoraussetzungen
Leistungskontrollen / Prüfungen
2 hand in presentations
- your log
- application / cv / ad
- scientific text

Lehrveranstaltungen / SWS
Blockseminar 120 h
40 h preparation / 40 h course / 40 h private study

Arbeitsaufwand
120 h

Modulnote

Lernziele / Kompetenzen

1. Communication
In this part, students learn about the meaning of communication in their everyday 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.

Inhalt

See above

Weitere Informationen

Unterrichtssprache: English

Literatur:
According to the topic
### Language Course

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>1-4</td>
<td>1 Semester</td>
<td>2 or 4</td>
<td>3 or 6</td>
<td></td>
</tr>
</tbody>
</table>

#### Modulverantwortliche/r
Dr. Peter Tischer, head of the Language Center

#### Dozent/inn/en
http://www.szsb.uni-saarland.de/mitarbeiter/

#### Zuordnung zum Curriculum
- Bachelor Informatik
- Master Informatik
For each language taught at the center, different levels are offered: beginner, intermediate and advanced level

Elective

#### Zulassungsvoraussetzungen
- 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.

#### Leistungkontrollen / Prüfungen
Usually exam at the end of the semester and regular attendance (at least 80 % of all classes).

#### Lehrveranstaltungen / SWS
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

#### Arbeitsaufwand
- 90 h = 30 h classes and 60 h private study
- 180 h = 60 h classes and 120 h private study

#### Modulnote
unbenotet

#### Lernziele / Kompetenzen
Language skills: grammar, vocabulary, conversation skills.

#### Inhalt
Depending on course
Weitere Informationen
Unterrichtssprache: German and taught language

Literatur:
Depending on course
Language Course - German Language Course for Beginners

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>1-4</td>
<td>1 Semester</td>
<td>2 or 4</td>
<td>3 or 6</td>
<td></td>
</tr>
</tbody>
</table>

Modulverantwortliche/r  
Dozent/inn/en  

Zuordnung zum Curriculum  
1.- 4. Semester / international Master students only  
Elective

Zulassungsvoraussetzungen  
none

Leistungskontrollen / Prüfungen  
Weekly assignments  
One presentation  
Exam at the end of the semester  
Regular attendance (at least 75% of all classes)

Lehrveranstaltungen / SWS  
Seminar 6 h of classes each week  
Groups of up to 20 students

Arbeitsaufwand  
270 h = 90 h of classes and 180 h private study

Modulnote  
unbenotet

Lernziele / Kompetenzen

Students should develop basic skills in

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

Inhalt

See above
Weitere Informationen

Unterrichtssprache: German

Literatur:
Depending on course
Language Course - German Language Course / all levels

<table>
<thead>
<tr>
<th>Studiensem.</th>
<th>Regelstudiensem.</th>
<th>Turnus</th>
<th>Dauer</th>
<th>SWS</th>
<th>ECTS-Punkte</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>1-4</td>
<td>1 Semester</td>
<td>2 or 4</td>
<td>3 or 6</td>
<td></td>
</tr>
</tbody>
</table>

Modulverantwortliche/r: NN
Dozent/inn/en: NN

Zuordnung zum Curriculum: 1.-3. Semester / courses are offered each semester

Zulassungsvoraussetzungen: Language test to assess the proficiency of the student

Leistungskontrollen / Prüfungen: Weekly assignments
One presentation
Exam at the end of the semester
Regular attendance (at least 75% of all classes)

Lehrveranstaltungen / SWS: Seminar 4 h of classes each week
Groups of up to 20 students

Arbeitsaufwand: 180 h = 60 h of classes and 120 h private study

Modulnote: unbenotet

Lernziele / Kompetenzen:
Students should develop basic skills in:

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

Inhalt:
See above

Weitere Informationen:
Unterrichtssprache: German

Literatur:
Depending on course