

Module Descriptions

Master Program Data Science and Artificial Intelligence

(01.10.2019)

Studiensem.	Module/Module Category	CP	SWS
Core Courses DSAI			
1 – 3	Artificial Intelligence	9	6
1 – 3	Information Retrieval and Data Mining	9	6
1 – 3	Database Systems	9	6
1 – 3	Machine Learning	9	6
1 – 3	Neural Networks: Theory and Application	9	6
1 – 3	Image Processing and Computer Vision	9	6
1 – 3	Automated Reasoning	9	6
Core Courses Computer Science			
1 – 3	Algorithms and Data Structures	9	6
1 – 3	Compiler Construction	9	6
1 – 3	Complexity Theory	9	6
1 – 3	Computer Algebra	9	6
1 – 3	Computer Graphics	9	6
1 - 3	Cryptography	9	6
1 – 3	Data Networks	9	6
1 – 3	Distributed Systems	9	6
1 – 3	Embedded Systems	9	6
1 – 3	Geometric Modeling	9	6
1 – 3	Human Computer Interaction	9	6
1 – 3	Introduction to Computational Logic	9	6
1 – 3	Operating Systems	9	6
1 - 3	Optimization	9	6

Studiensem.	Module/Module Category	CP	SWS
1 – 3	Semantics	9	6
1 – 3	Software Engineering	9	6
1 – 3	Telecommunications 1	9	6
1 – 3	Verification	9	6
Advanced Courses DSAI			
1 - 3	Ethics for Nerds	9	6
1 - 3	Security	9	6
1 - 3	Stochastik 1	6	4
1 - 3	Stochastik 2	6	4
1 - 3	Mathematische Statistik	9	6
1 - 3	AI Planning	9	6
Module Category			
1 - 3	Seminar	6	4
Module			
3	Master Seminar	6	4
1 - 3	Master Thesis	6	4
Freely chosen points (elective courses)			

Artificial Intelligence					CS 556 / AI
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 – 3	1 - 2	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Jörg Hoffmann
Lecturer	Prof. Dr. Jörg Hoffmann, Prof. Dr. Jana Köhler
Level of the unit / mandatory or not	Master DSAI, Core Lecture DSAI, Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials • Solving of weekly assignments • Passing the final written exam • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 30 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in written tests, tutor groups, and the final exam. Details will be announced by the lecturer at the beginning of the course.

Aims / Competences to be developed

Knowledge about basic methods in Artificial Intelligence

Teaching language: English

Content

Problem-solving:

- Uninformed- and informed search procedures
- Adversarial search

Knowledge and reasoning:

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

Planning:

- STRIPS formalism and complexity
- Delete relaxation heuristics

Probabilistic reasoning:

- Basic probabilistic methods
- Bayesian networks

Additional Information

Teaching language: English

Literature:

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

Information Retrieval and Data Mining					CS 555 / IRDM
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 – 3	1 - 2	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Gerhard Weikum
Lecturer	Prof. Dr. Gerhard Weikum
Level of the unit / mandatory or not	Master DSAI, Core Lecture DSAI, Mandatory Elective
Entrance requirements	Good knowledge of undergraduate mathematics (linear algebra, probability theory) and basic algorithms.
Assessment / Exams	<ul style="list-style-type: none"> • Regular attendance of classes and tutor groups • Presentation of solutions in tutor groups • Passing 2 of 3 written tests (after each third of the semester) • Passing the final exam (at the end of the semester)
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in written tests, tutor groups, and the final exam. Details will be announced on the course web site.

Aims / Competences to be developed

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

Content

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

Additional Information

Teaching language: English

Literature:
will be announced on the course web site.

Database Systems					CS 553 / DBS
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 – 3	1 - 2	At least once every two years	1 Semester	6	9

Responsible Lecturer Prof. Dr. Jens Dittrich
Lecturer Prof. Dr. Jens Dittrich
Level of the unit / mandatory or not Master DSAI, Core Lecture DSAI, Mandatory Elective

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

For graduate students:

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

Assessment / Exams

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

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

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

Course Typ / weekly hours	Lecture 4 h (weekly; this class may be run as a flipped classroom, i.e. 2 hours may be replaced by self-study of videos/papers; the other 2 hours may be used to run a group exercise supervised by the professor called "the LAB") Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade	Will be determined based on project, midterm and best of endterm and reexam.

Aims / Competences to be developed

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

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

Content

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

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

- write-optimized system concepts (OLTP, streaming data)
- management of geographical data (GIS, google maps and similar tools)
- main-memory techniques

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

Machine Learning					ML
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 – 3	1 - 2	At least once every two years	1 Semester	6	9

Responsible Lecturer N. N.

Lecturer N. N.

Level of the unit / mandatory or not Master DSAI, Core Lecture DSAI, Mandatory Elective

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

Assessment / Exams

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

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

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

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

Aims / Competences to be developed

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

Content

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

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

Neural Networks: Theory and Application					NNIA
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
4 – 5	4 - 5	At least once every two years	1 Semester	6	9

Modulverantwortliche/r	Prof. Dr. Dietrich Klakow
Dozent/inn/en	Prof. Dr. Dietrich Klakow
Zuordnung zum Curriculum	Master DSAI, Core Lecture DSAI, Mandatory Elective
Zulassungsvoraussetzungen	Mathematik für Informatiker 1 – 3 or comparable; good programming skills
Leistungskontrollen / Prüfungen	Written Exam
Lehrveranstaltungen / SWS	Lecture 2 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students Project work 2 h (weekly)
Arbeitsaufwand	270 h = 90 h of classes and 180 h individual work
Modulnote	Written exam and graded projects. Exact details will be announced in the first lecture

Lernziele / Kompetenzen

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

Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:

Ian Goodfellow and Yoshua Bengio and Aaron Courville

Deep Learning

MIT Press, 2016

<http://www.deeplearningbook.org>

Image Processing and Computer Vision					CS 572 / IPCV
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 – 3	1 - 2	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Joachim Weickert
Lecturer	Prof. Dr. Joachim Weickert
Level of the unit / mandatory or not	Master DSAI, Core Lecture DSAI, Mandatory Elective

Entrance requirements For graduate students: none

Assessment / Exams

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

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

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

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

Aims / Competences to be developed

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

Content

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

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

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

Automated Reasoning					CS 571 / AR
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 – 3	1 - 2	At least once every two years	1 Semester	6	9

Responsible Lecturer Prof. Dr. Christoph Weidenbach

Lecturer Prof. Dr. Christoph Weidenbach

Level of the unit / mandatory or not Master DSAI, Core Lecture DSAI, Mandatory Elective

Entrance requirements CS 575 ICL

Assessment / Exams

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

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

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

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

Aims / Competences to be developed

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

Content

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

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Dr. Kurt Mehlhorn
Lecturer	Prof. Dr. Kurt Mehlhorn, Prof. Dr. Raimund Seidel
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	For graduate students: C, C++, Java
Assessment / Exams	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials • Passing the midterm and the final exam • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.
Aims / Competences to be developed	The students know standard algorithms for typical problems in the areas graphs, computational geometry, strings and optimization. Additionally more they master a number of methods and data-structures to develop efficient algorithms and analyze their running times.

Content

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

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Dr. Sebastian Hack
Lecturer	Prof. Dr. Sebastian Hack
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials • Written exam at the end of the course, theoretical exercises, and compiler-laboratory project. • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.
Aims / Competences to be developed	

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

Content

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

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

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

Aims / Competences to be developed

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

Content

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

Additional Information

Teaching language: English

Literature:

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

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

Responsible Lecturer	Prof. Dr. Frank-Olaf Schreyer
Lecturer	Prof. Dr. Frank-Olaf Schreyer
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials • Solving the exercises, passing the midterm and the final exam. • Grade: 20% exercises, 30% midterm, 50% final exam.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.
Aims / Competences to be developed	Solving problems occurring in computer algebra praxis The theory behind algorithms

Content

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

- integer and modular arithmetics, prime number tests
- 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

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Dr. Philipp Slusallek
Lecturer	Prof. Dr. Philipp Slusallek
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	<ul style="list-style-type: none"> • Successful completion of at least 50% of the exercises • Successful participation in rendering competition • Final written exam <p>Final grade determined by result of the exam and the rendering competition</p> <p>A re-exam takes place during the last two weeks before the start of lectures in the following semester.</p>
Course Typ / weekly hours	<p>Lecture 4 h (weekly)</p> <p>Tutorial 2 h (weekly)</p> <p>Tutorials in groups of up to 20 students</p>
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.
Aims / Competences to be developed	<p>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 use it in a high-quality rendering.</p>

Content

- 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

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Dr. Michael Backes
Lecturer	Prof. Dr. Michael Backes, Prof. Dr. Markus Bläser, Dr. Nico Döttling
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	For graduate students: Basic knowledge in theoretical computer science required, background knowledge in number theory and complexity theory helpful
Assessment / Exams	<ul style="list-style-type: none"> • Oral / written exam (depending on the number of students) • A re-exam is normally provided (as written or oral examination).
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.

Aims / Competences to be developed

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

Content

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

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Dr. Holger Hermanns
Lecturer	Prof. Dr. Holger Hermanns
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials • Qualification for final exam through mini quizzes during classes • Possibility to get bonus points through excellent homework • Final exam • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.
Aims / Competences to be developed	<p>After taking the course students have</p> <ul style="list-style-type: none"> • a thorough knowledge regarding the basic principles of communication networks, • the fundamentals of protocols and concepts of protocol, • Insights into fundamental motivations of different pragmatics of current network solutions, • Introduction to practical aspects of data networks focusing on internet protocol hierarchies

Content

Introduction and overview

Cross section:

- Stochastic Processes, Markov models,
- Fundamentals of data network performance assessment
- Principles of reliable data transfer
- Protocols and their elementary parts
- Graphs and Graphalgorithms (maximal flow, spanning tree)
- Application layer:
- Services and protocols
- FTP, Telnet
- Electronic Mail (Basics and Principles, SMTP, POP3, ..)
- World Wide Web (History, HTTP, HTML)
- Transport Layer:
- Services and protocols
- Addressing
- Connections and ports
- Flow control
- QoS
- Transport Protocols (UDP, TCP, SCTP, Ports)
- Network layer:
- Services and protocols
- Routing algorithms
- Congestion Control
- Addressing
- Internet protocol (IP)
- Data link layer:
- Services and protocols
- Medium access protocols: Aloha, CSMA (-CD/CA), Token passing
- Error correcting codes
- Flow control
- Applications: LAN, Ethernet, Token Architectures, WLAN, ATM
- Physical layer
- Peer-to-Peer and Ad-hoc Networking Principles

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Peter Druschel, Ph.D.
Lecturer	Prof. Peter Druschel, Ph.D. Allen Clement, Ph.D.
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	Operating systems or concurrent programming.
Assessment / Exams	<ul style="list-style-type: none"> • Regular attendance at classes and tutorials. • Successful completion of a course project in teams of 2 students. (Project assignments due approximately every 2 weeks.) • Passing grade on 2 out of 3 written exams: midterm, final exam, and a re-exam that takes place during the last two weeks before the start of lectures in the following semester. • Final course grade: 50% project, 50% best 2 out of 3 exams.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly)
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.
Aims / Competences to be developed	Introduction to the principles, design, and implementation of distributed systems
Content	<ul style="list-style-type: none"> - Communication: Remote procedure call, distributed objects, event notification, content dissemination, group communication, epidemic protocols. - Distributed storage systems: Caching, logging, recovery, leases. - Naming. Scalable name resolution. - Synchronization: Clock synchronization, logical clocks, vector clocks, distributed snapshots. - Fault tolerance: Replication protocols, consistency models, consistency versus availability trade-offs, state machine replication, consensus, Paxos, PBFT. - Peer-to-peer systems: consistent hashing, self-organization, incentives, distributed hash tables, content distribution networks. - Data centers. Architecture and infrastructure, distributed programming, energy efficiency.

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Bernd Finkbeiner, Ph.D
Lecturer	Prof. Bernd Finkbeiner, Ph.D
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	
Assessment / Exams	<ul style="list-style-type: none"> • Written exam at the end of the course. • Demonstration of the implemented system. • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) The course is accompanied by a laboratory project, in which a non-trivial embedded system has to be realized.
Total workload	270 h = 90 h classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.
Aims / Competences to be developed	
	The students should learn methods for the design, the implementation, and the validation of safety-critical embedded systems.

Content

Embedded Computer Systems are components of a technical system, e.g. an air plane, a car, a household machine, a production facility. They control some part of this system, often called the plant, e.g. the airbag controller in a car controls one or several airbags. Controlling means obtaining sensor values and computing values of actuator signals and sending them.

Most software taught in programming courses is transformational, i.e. it is started on some input, computes the corresponding output and terminates. Embedded software is reactive, i.e. it is continuously active waiting for signals from the plant and issuing signals to the plant.

Many embedded systems control safety-critical systems, i.e. malfunctioning of the system will in general cause severe damage. In addition, many have to satisfy real-time requirements, i.e. their reactions to input have to be produced within fixed deadlines.

According to recent statistics, more than 99% of all processors are embedded. Processors in the ubiquitous PC are a negligible minority. Embedded systems have a great economical impact as most innovations in domains like avionics, automotive are connected to advances in computer control. On the other hand, failures in the design of such systems may have disastrous consequences for the functioning of the overall system. Therefore, formal specification techniques and automatic synthesis of software are used more than in other domains.

The course will cover most aspects of the design and implementation of embedded systems, e.g. specification mechanisms, embedded hardware, operating systems, scheduling, validation methods.

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Dr. Hans-Peter Seidel
Lecturer	Prof. Dr. Hans-Peter Seidel, Prof. Dr. Philipp Slusallek
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	<ul style="list-style-type: none"> • Regular attendance and participation. • Weekly Assignments (10% bonus towards the course grade; bonus points can only improve the grade; they do not affect passing) • Passing the written exams (mid-term and final exam). • The mid-term and the final exam count for 50% each, but 10% bonus from assignments will be added. • A re-exam takes place at the end of the semester break or early in the next semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students (theory) Practical assignments in groups of 3 students (practice) Tutorials consists of a mix of theoretical + practical assignments.
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be based on the performance in exams, exercises and practical tasks. The detailed terms will be announced by the module coordinator.
Aims / Competences to be developed	Gaining knowledge of the theoretical aspect of geometric modelling problems, and the practical solutions used for modelling and manipulating curves and surfaces on a computer. From a broader perspective: Learning how to represent and interact with geometric models in a discretized, digital form (geometric representations by functions and samples; design of linear function spaces; finding “good” functions with respect to a geometric modelling task in such spaces).

Content

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

Additional Information

Teaching language:English

Human Computer Interaction, Core Course					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	1	At least once every two years	1 Semester	6	9

Modulverantwortlicher	Prof. Dr. Jürgen Steimle
Dozent	Prof. Dr. Jürgen Steimle
Zuordnung zum Curriculum	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Zulassungsvoraussetzungen	For BsC students in Computer Science: Programmierung 1 + 2 For graduate students in Computer Science and Media Informatics : none
Leistungskontrollen / Prüfungen	Regular attendance of classes and tutorials Successful completion of exercises and course project Final exam A re-exam takes place (as written or oral examination).
Lehrveranstaltungen /SWS	Lecture 4 h (weekly) Tutorial 2 h (weekly)
Arbeitsaufwand	270 h = 90 h of classes and 180 h private study
Modulnote	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.

Lernziele / Kompetenzen

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

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

Inhalt

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

Weitere Informationen

Unterrichtssprache: Englisch

Literaturhinweise:

Bekanntgabe jeweils vor Beginn der Vorlesung auf der Vorlesungsseite im Internet

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

Responsible Lecturer	Prof. Dr. Gert Smolka
Lecturer	Prof. Dr. Gert Smolka
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	
Assessment / Exams	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials. • Passing the midterm and the final exam.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.
Aims / Competences to be developed	<ul style="list-style-type: none"> • structure of logic languages based on type theory • distinction notation / syntax / semantics • structure and formal representation of mathematical statements • structure and formal representation of proofs (equational and natural deduction) • solving Boolean equations • proving formulas with quantifiers • implementing syntax and deduction

Content

Type Theory

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

Predicate Logic (higher-order)

- quantifier axioms
- natural deduction
- prenex and Skolem forms

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Peter Druschel, Ph.D.
Lecturer	Prof. Peter Druschel, Ph.D. Björn Brandenburg, Ph.D.
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective

Entrance requirements

Assessment / Exams	Regular attendance at classes and tutorials Successful completion of a course project in teams of 2 students Passing 2 written exams (midterm and final exam) A re-exam takes place during the last two weeks before the start of lectures in the following semester.
---------------------------	--

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

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

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

Aims / Competences to be developed

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

Content

Process management:

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

Memory management:

- Dynamic storage allocation
- Sharing main memory
- Virtual memory

I/O management:

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

Virtual machines

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Dr. Kurt Mehlhorn
Lecturer	Prof. Dr. Kurt Mehlhorn
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	<ul style="list-style-type: none"> • 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.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course

Aims / Competences to be developed

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

Content

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

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Dr. Gert Smolka
Lecturer	Prof. Dr. Gert Smolka
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	For graduate students: core lecture Introduction to Computational Logic
Assessment / Exams	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials. • Passing the midterm and the final exam
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.

Aims / Competences to be developed

Understanding of

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

Content

Theory of programming languages, in particular:

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

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Dr. Sven Apel
Lecturer	Prof. Dr. Sven Apel
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	<p>Basic knowledge in programming concepts (as taught in courses such as Programming 2 and Software-Praktikum)</p> <p>The number of students admitted to participate in this course is limited. The number of attendees will be announced on the course website several weeks before course starts. Students interested in this course need to sign up on the course's website before the course starts within a registration period announced on the course website. Before start, the participants will be determined from the students registered on the course website by lottery. Admitted students will be notified before the course starts.</p>
Assessment / Exams	<ul style="list-style-type: none"> • Successful project completion (including weekly progress reports and deliverables, such as requirements, design, implementation) • Successful project demonstration • Regular attendance of classes • Passing the final exam
Course Typ / weekly hours	<p>Lecture 2 h (weekly)</p> <p>Project 4 h (weekly)</p> <p>Project work in teams of 4–7 students</p>
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.
Aims / Competences to be developed	The students know and apply modern software development techniques.

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

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

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

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

They apply these techniques in a project in small teams.

Lecture Contents

- Requirements Engineering
- Responsibility-Driven Design
- Object-Oriented Software Design
- Software Architecture
- Coding Techniques and Guidelines
- Software Maintenance and Evolution
- Software Testing

Additional Information

Teaching language: English

Literature:

- Software Engineering. I. Sommerville. Addison-Wesley, 2004.
- Software Engineering: A Practitioner's Approach. R. Pressman. McGraw Hill Text, 2001.
- Using UML: Software Engineering with Objects and Components. P. Stevens, R. Pooley. Addison-Wesley, 1999.
- UML Distilled. M. Fowler, K. Scott. Addison-Wesley, 2000.
- Objects, Components and Frameworks with UML, D. D'Souza, A. Wills. Addison-Wesley, 1999.
- Designing Object-Oriented Software. R. Wirfs-Brock, B. Wilkerson, L. Wiener. Prentice Hall, 1990.
- Design Patterns. Elements of Reusable Object-Oriented Software. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Addison Wesley, 1995.
- Pattern-Oriented Software Architecture: A System of Patterns. F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. Wiley, 1996.
- Software Architecture: Perspectives on an Emerging Discipline. M. Shaw, D. Garlan. Prentice-Hall, 1996.
- Refactoring: Improving the Design of Existing Code. M. Fowler, K. Beck, W. Opdyke. Addison-Wesley, 1999.
- Software Testing and Analysis: Process, Principles and Techniques. M. Pezze. Wiley. 2007.

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

Responsible Lecturer	Prof. Dr.-Ing. Thorsten Herfet
Lecturer	Prof. Dr.-Ing. Thorsten Herfet
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	The lecture requires a solid foundation of mathematics (differential and integral calculus) and probability theory. The course will, however, refresh those areas indispensably necessary for telecommunications and potential intensification courses and by this open this potential field of intensification to everyone of you.
Assessment / Exams	Regular attendance of classes and tutorials Passing the final exam in the 2nd week after the end of courses. Eligibility: Weekly exercises / task sheets, grouped into two blocks corresponding to first and second half of the lecture. Students must provide min. 50% grade in each of the two blocks to be eligible for the exam.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Final exam mark
Aims / Competences to be developed	Digital Signal Transmission and Signal Processing refreshes the foundation laid in "Signals and Systems" [Modulkennung]. Including, however, the respective basics so that the various facets of the introductory study period (Bachelor in Computer Science, Vordiplom Computer- und Kommunikationstechnik, Elektrotechnik or Mechatronik) and the potential main study period (Master in Computer Science, Diplom-Ingenieur Computer- und Kommunikationstechnik or Mechatronik) will be paid respect to.

Content

As the basic principle, the course will give an introduction into the various building blocks that modern 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.

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

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

Responsible Lecturer	Prof. Dr. Holger Hermanns
Lecturer	Prof. Dr. Holger Hermanns, Prof. Bernd Finkbeiner, Ph.D
Level of the unit / mandatory or not	Master DSAI, Core Lecture Computer Science, Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials • Passing the final exam • A re-exam takes place during the last two weeks before the start of lectures in the following semester.
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.

Aims / Competences to be developed

The students become familiar with the standard methods in computer-aided verification. They understand the theoretical foundations and are able to assess the advantages and disadvantages of different methods for a specific verification project.

The students gain first experience with manual correctness proofs and with the use of verification tools.

Content

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

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

Security					CS 559 / SEC
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	1 - 2	At least once every two years	1 Semester	6	9

Responsible Lecturer	Prof. Dr. Michael Backes
Lecturer	Prof. Dr. Michael Backes, Prof. Dr. Cas Cremers
Level of the unit / mandatory or not	Master DSAI, Advanced Lecture DSAI, Mandatory Elective
Entrance requirements	For graduate students: none
Assessment / Exams	<ul style="list-style-type: none"> • Regular attendance of classes and tutorials • Passing the final exam • A re-exam is normally provided (as written or oral examination).
Course Typ / weekly hours	Lecture 4 h (weekly) Tutorial 2 h (weekly) Tutorials in groups of up to 20 students
Total workload	270 h = 90 h of classes and 180 h private study
Grade of the module	Will be determined by the performance in exams, tutor groups, and practical tasks. Details will be announced by the lecturer at the beginning of the course.
Aims / Competences to be developed	

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

Content

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

Additional Information

Teaching language: English

Literature:

Will be announced on the course website

AI Planning					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 – 3	1 - 2	WS	1 Semester	6	9

Responsible Lecturer

Prof. Dr Jörg Hoffmann

Lecturer

Prof. Dr. Jörg Hoffmann

Level of the unit / mandatory or not

Master DSAI, Advanced Lecture DSAI,
Mandatory Elective

Entrance requirements

For graduate students: none

Assessment / Exams

Regular attendance of classes and tutorial
Paper as well as programming exercises for exam
qualification
Final exam
A re-exam takes place before the start of lectures in the
following semester.

Course Typ / weekly hours

Lecture 4 h (weekly)
Tutorial 2 h (weekly)

Total workload

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

Grade of the module

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

Aims / Competences to be developed

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

Content

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

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

Additional Information

Unterrichtssprache: Englisch

Literaturhinweise:

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

Seminar Changing Topics					CS 500
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 - 3	2	jedes Semester	1 Semester	3	7

Responsible Lecturer	Dean of studies and relevant Professor
Lecturer	Professors of the Department
Level of the unit / mandatory or not	Master DSAI, Mandatory Elective
Entrance requirements	Basic knowledge in the field of computer science under focus in the respective seminar.
Assessment / Exams	<ul style="list-style-type: none"> • Contributions to discussions • Thematic talk • Written elaboration • Final oral examination on the entire scientific area spanned by the seminar
Course Typ / weekly hours	Seminar 3 h (weekly) / groups of up to 20 students
Total workload	210 h = 45 h classes und 165 h private study
Grade of the module	The modalities of the grading will be determined by the responsible professor

Aims / Competences to be developed

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

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

Content

Practical exercising of

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

Specific focus according to the individual topic of the seminar.

Typical course progression:

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

Oral exam on entire scientific area spanned by the seminar

Additional Information

Teaching language: English

Literatur:

According to the topic

Master Seminar					CS 890
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
3	3	jedes Semester	1 Semester	5	12

Responsible Lecturer

Dean of Studies

Lecturer

Professors of the department

Level of the unit / mandatory or not

Master DSAI

Entrance requirements

Acquisition of at least 30 CP

Assessment / Exams

- Presentation of a scientific article of adequate depth in the reading group
- Active participation in the discussion in the reading group
- Presentation of the planned thesis topic followed by a plenary discussion
- Written description of the topic of the Master thesis

Course Typ / weekly hours

Seminar 3h per week (about 15 members)

Total workload

- 360 h
- Hands-on training 2h per week (15 students)
 - Contact with supervisor 1h per week
 - Self-study 20h per week

Grade of the module

graded

Aims / Competences to be developed

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

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

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

Content

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

Additional Information

Teaching language: English

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

Responsible Lecturer Professors of the Department

Lecturer Professors of the Department

Level of the unit / mandatory or not Master DSAI

Entrance requirements Master Seminar

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

Course Typ / weekly hours

Total workload 900 h = 50 h contact hours, 850 h private studies

Grade of the module graded

Aims / Competences to be developed

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

Content

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

Additional Information

Teaching language: English

Literature:
According to the topic

Wahlpflicht					
Studiensem.	Regelstudiensem.	Turnus	Dauer	SWS	ECTS-Punkte
1 – 3	2 - 3	Jedes Semester	1 Semester	variabel	variabel

Modulverantwortliche/r

Studiendekan der Fakultät Mathematik und Informatik
bzw. Studienbeauftragter der Informatik

Dozent/inn/en

Zuordnung zum Curriculum

Master DSAI, Wahlpflicht

Zulassungsvoraussetzungen

Keine

Leistungskontrollen / Prüfungen

Lehrveranstaltungen / SWS

Wählbare Veranstaltungen im Umfang von mind. 7 CP aus folgenden Bereichen:
 Soft Skills Veranstaltungen laut Kursangebot, z. B.:
Tutortätigkeit, 4 CP
Soft Skills Seminar, 4 CP
Sprachkurse, 3 oder 6 CP
Industrie-Praktikum
Beliebig wählbare Module des Bachelor-Studiengangs DSAI und Informatik

Arbeitsaufwand

Arbeitsaufwand: insgesamt 240 Stunden

Modulnote

Das Modul ist insgesamt bestanden, wenn die Prüfungsleistung bestanden wurde. Die Leistungen sind unbenotet.

Lernziele / Kompetenzen

- Veranstaltungen des Fachbereichs Informatik:

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

- Soft Skills Veranstaltungen:

- Tutoren lernen, wie Lehrveranstaltungen organisiert werden und welche methodischen Ziele dabei verfolgt werden. Sie lernen, komplexe fachliche Inhalte sowohl in einer größeren Gruppe (Übungsgruppe) als auch in individuellen Beratungsgesprächen zu vermitteln.
- Präsentationstechniken, wissenschaftliche Recherche, Projektmanagement
- Erlernen verschiedener Fremdsprachen in Wort und Schrift

Inhalte

- Veranstaltungen des Fachbereichs Informatik (Stammvorlesungen und Vertiefungsvorlesungen):

Der Inhalt variiert nach belegtem Themenschwerpunkt. Das Kursangebot kann variieren und orientiert sich an dem Vorlesungsangebot des Fachbereichs und spiegelt die Forschungsthemen der

Saarbrücker Informatik wieder. In den Veranstaltungen werden zentrale wissenschaftliche Fragestellungen der Kerngebiete der Informatik vorgestellt und behandelt.

Weitere Informationen

Die Unterrichtssprache ist deutsch oder englisch und wird zu Beginn der Veranstaltung bekannt gegeben.