UNIVERSITÄT
DES
SAARLANDES

Fakultät für Mathematik und Informatik

Modulhandbuch

Entrepreneurial Cybersecurity MSc

4. Februar 2024

Liste der Modulbereiche und Module

Modulbereich 1

Gründungsprojekte

Gründungsprojekt 1 / 2 / 3

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
2 / 3 / 4	**4**	**jedes Semester**	**je 1 Semester**	**je 10**	**je 15**

Modulverantwortliche/r Prof. Dr. Andreas Zeller

Dozent/inn/en Professor/inn/en und Forscher/inn/en der Cybersicherheit

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen Die Prüfungsleistung besteht aus drei Teilen, die am Ende des Projektes einzureichen und vorzustellen sind:

1. Sales Pitch:
   - Gründungsprojekt 1: Mit Prüfer/inn/en
   - Gründungsprojekt 2: Mit Prüfer/inn/en und Gästen
   - Gründungsprojekt 3: Mit Prüfer/inn/en sowie Investoren und Förderern
2. Demonstration einer prototypischen Lösung:
   - Gründungsprojekt 1: Erster Prototyp, um Machbarkeit zu demonstrieren
   - Gründungsprojekt 2: Zweiter Prototyp mit Demonstration der Bedienbarkeit (Horizontaler Prototyp) und Machbarkeit (Vertikaler Prototyp)
   - Gründungsprojekt 3: Vollständiger Demonstrator der Geschäftsidee
3. Schriftliche Unterlagen:
   - Gründungsprojekt 1: Beschreibung der Geschäftsidee und des Teams; Analyse des Marktes und der Zielgruppe
   - Gründungsprojekt 2: wie 1, zusätzlich Analyse von Marketing und Vertrieb;
   - Gründungsprojekt 3: wie 2; zusätzlich mit Unternehmensorganisation und Finanzplanung.

Lehrveranstaltungen / SWS `10 SWS Praktikum`

Arbeitsaufwand

```
  30 h Präsenzstudium
+ 420 h Eigenstudium
= 450 h (= 15 ECTS)
```

Modulnote Wird bestimmt aus Qualität des Sales Pitches, der Demonstration der Prototypen, und der schriftlichen Ausarbeitungen. Details werden vom Dozenten/von der Dozentin zu Beginn der Vorlesung bekanntgegeben.

Sprache Deutsch oder Englisch

Lernziele / Kompetenzen

Die Studierenden lernen, eine Geschäftsidee aus aktuellen Forschungsthemen der Cybersecurity zu entwickeln, sie marktorientiert zu verfeinern und sie prototypisch umzusetzen mit dem Ziel einer Proof-of-Concept-Demonstration, die zu einer Unternehmensgründung führen kann. Sie erlernen systematisches Planen von technischen wie finanziellen Aspekten, nachhaltiges Entwickeln anspruchsvoller Konzepte, sowie verantwortungsbewusstes Arbeiten im Team.

Inhalt

In diesem Praktikum, das sich über drei Semester erstreckt, erarbeiten Studierende eine Geschäftsidee aus dem Bereich der Cybersecurity, mit dem Ziel, anschließend / begleitend ein darauf aufbauendes Unternehmen gründen zu können. Die Geschäftsidee soll aktuelle Forschungsthemen marktfähig umsetzen; sie soll während des Projekts von den Studierenden unter Mitwirkung von Mentoren aus Wissenschaft und Wirtschaft entwickelt und verfeinert werden.

Literaturhinweise

keine

Weitere Informationen

Im Rahmen des Gründungsprojekts werden Gastvorlesungen mit Gründer/inn/en, Förderer/inne/n, und Berater/inn/en mit Gesprächen im kleinen Kreis angeboten. Details gibt der Dozent/die Dozentin bekannt.

Studierende können sich zu Beginn eines Semesters jeweils als Gruppe von 2–4 Personen für das Gründungsprojekt anmelden. Der Dozent/die Dozentin kann auf Antrag Einzelpersonen oder größere Gruppen zulassen.

Jeder Gruppe wird ein/e Mentor/in aus der Gruppe der Betreuer/inn/en zugeordnet, die der Gruppe für technisch-wissenschaftliche Beratung und weitere Anregungen im Umfang von bis zu 2h/Woche zur Verfügung steht. Es ist möglich, dass ein/e Mentor/in als Gründungspartner/in, Investor/in oder wissenschaftliche/r Berater/in des zu gründenden Unternehmens agiert.

Im laufenden Gründungsprojekt kann der Dozent/die Dozentin auf Antrag Studierende einer anderen Gruppe zuweisen oder sie zu Einzelteilnehmer/inn/en machen; die Studierende/n und beteiligten Gruppen sind dabei zu hören. Ein Wechsel eines/einer Studierenden mehr als einmal pro Semester oder später als acht Wochen nach Vorlesungsbeginn ist nur in Fällen möglich, die außerhalb des Einflussbereichs des/der Studierenden liegen.

Es ist erwünscht, dass Projekte und Gruppenzusammensetzungen des vergangenen Semesters im Gründungsprojekt des Folgesemesters fortgeführt werden. Es wird empfohlen, Erfindungen im Rahmen des Gründungsprojekts frühzeitig zu melden. Details hierzu gibt der Dozent/die Dozentin bekannt.

Für das Schlichten von Streitigkeiten während des Gründungsprojektes benennt der Dozent/die Dozentin eine Ombudsperson aus dem Bereich der Professor/inn/en. Details hierzu gibt der Dozent/die Dozentin bekannt.

Modulbereich 2

Stammvorlesungen Cybersecurity

Cryptography — Crypto

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Dr. Nico Döttling

Dozent/inn/en Prof. Dr. Cas Cremers
Dr. Nico Döttling
Dr. Antoine Joux
Dr. Lucjan Hanzlik
Dr. Julian Loss

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

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

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

Sprache English

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

Literaturhinweise

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

Data Networks DN

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr.-Ing. Holger Hermanns

Dozent/inn/en Prof. Dr.-Ing. Holger Hermanns
Prof. Dr. Anja Feldmann

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

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

Arbeitsaufwand

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

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

Sprache English

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

Literaturhinweise

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

Embedded Systems ES

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	4	at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Bernd Finkbeiner, Ph.D

Dozent/inn/en Prof. Bernd Finkbeiner, Ph.D
Prof. Dr. Martina Maggio

Zulassungsvoraussetzungen keine

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

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

The course is accompanied by a laboratory project, in which a non-trivial embedded system has to be realized.

Arbeitsaufwand

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

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

Sprache English

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 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
markdownRendererInterblockSeparator 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.

Literaturhinweise

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

Machine Learning ML

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	4	at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Isabel Valera

Dozent/inn/en Prof. Dr. Isabel Valera

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 have to be obtained in order to qualify for the exam.
- Passing 1 out of 2 exams (final, re-exam).

Lehrveranstaltungen / SWS

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

Arbeitsaufwand

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

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

Sprache English

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

Literaturhinweise

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

Operating Systems OS

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

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

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

Zulassungsvoraussetzungen For graduate students: none

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

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

Arbeitsaufwand

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

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

Sprache English

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

Literaturhinweise

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

Security Sec

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Michael Backes

Dozent/inn/en Prof. Dr. Michael Backes
Prof. Dr. Cas Cremers

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

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

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.

Sprache English

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

Literaturhinweise

Will be announced on the course website

Software Engineering SE

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Sven Apel

Dozent/inn/en Prof. Dr. Sven Apel

Zulassungsvoraussetzungen

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

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

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

Lehrveranstaltungen / SWS

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

Arbeitsaufwand

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

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

Sprache English

Lernziele / Kompetenzen

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

Inhalt

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

Literaturhinweise

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

Verification Veri

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	4	at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr.-Ing. Holger Hermanns

Dozent/inn/en Prof. Dr.-Ing. Holger Hermanns
Prof. Bernd Finkbeiner, Ph.D

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen
- 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.

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

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

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

Sprache English

Lernziele / Kompetenzen

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

Literaturhinweise

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

Modulbereich 3

Stammvorlesungen Informatik

Algorithms and Data Structures — AlgoDat

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Kurt Mehlhorn

Dozent/inn/en Prof. Dr. Raimund Seidel
Prof. Dr. Kurt Mehlhorn

Zulassungsvoraussetzungen For graduate students: C, C++, Java

Leistungskontrollen / Prüfungen
- 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.

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

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

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

Sprache English

Lernziele / Kompetenzen

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

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.

Literaturhinweise

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

Artificial Intelligence AI

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Jörg Hoffmann

Dozent/inn/en Prof. Dr. Jörg Hoffmann

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

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

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

Arbeitsaufwand

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

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

Sprache English

Lernziele / Kompetenzen

Knowledge about basic methods in Artificial Intelligence

Inhalt

Search:

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

Planning:

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

Markov decision processes:

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

Games:

- Adversarial search
- Learning from self-play

Literaturhinweise

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

Automated Reasoning AR

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Christoph Weidenbach

Dozent/inn/en Prof. Dr. Christoph Weidenbach

Zulassungsvoraussetzungen *Introduction to Computational Logic*

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

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

Arbeitsaufwand

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

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

Sprache English

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

Literaturhinweise

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

Compiler Construction CC

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Sebastian Hack

Dozent/inn/en Prof. Dr. Sebastian Hack

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

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

Arbeitsaufwand

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

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

Sprache English

Lernziele / Kompetenzen

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

Inhalt

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

Literaturhinweise

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

Complexity Theory CT

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Markus Bläser

Dozent/inn/en Prof. Dr. Raimund Seidel
Prof. Dr. Markus Bläser

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

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

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

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

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

Sprache English

Lernziele / Kompetenzen

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.

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-Szelepcsenyi theorem, polynomial time hierarchy, relativization, parity and the polynomial methods, Valiant-Vazirani theorem, counting problems and classes, Toda's theorem, probabilistic computations, isolation lemma and parallel algorithms for matching, circuit identity testing, graph isomorphism and interactive proofs.

Literaturhinweise

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

Computer Algebra CA

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	4	at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Frank-Olaf Schreyer

Dozent/inn/en Prof. Dr. Frank-Olaf Schreyer

Zulassungsvoraussetzungen For graduate students: none

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

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

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

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

Sprache English

Lernziele / Kompetenzen

Solving problems occuring in computer algebra praxis
The theory behind algorithms

Inhalt

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

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

Literaturhinweise

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

Computer Architecture CAr

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

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

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

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

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

Arbeitsaufwand

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

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

Sprache English

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

Literaturhinweise

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

Computer Graphics CG

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Philipp Slusallek

Dozent/inn/en Prof. Dr. Philipp Slusallek

Zulassungsvoraussetzungen Solid knowledge of linear algebra is recommended.

Leistungskontrollen / Prüfungen

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

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

Lehrveranstaltungen / SWS

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

Arbeitsaufwand

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

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

Sprache English

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 has some focus on image synthesis or rendering. The first part of the course uses ray tracing as a driving applications to discuss core topics of computer graphics, from vector algebra all the way to sampling theory, the human visual system, sampling theory, and spline curves and surfaces. A second part then uses rasterization approach as a driving example, introducing the camera transformation, clipping, the OpenGL API and shading langue, plus advanced techniques.

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

Inhalt

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

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

Literaturhinweise

Will be announced in the lecture.

Continious Optimization — OPT

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Peter Ochs

Dozent/inn/en Prof. Dr. Peter Ochs

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

Leistungskontrollen / Prüfungen

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

Lehrveranstaltungen / SWS

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

Arbeitsaufwand

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

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

Sprache English

Lernziele / Kompetenzen

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

Inhalt

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

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

Literaturhinweise

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

Convex Analysis and Optimization — CAO

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Peter Ochs

Dozent/inn/en Prof. Dr. Peter Ochs

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

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

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

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

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

Sprache English

Lernziele / Kompetenzen

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

Inhalt

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

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

Literaturhinweise

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

Database Systems DBS

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Jens Dittrich

Dozent/inn/en Prof. Dr. Jens Dittrich

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

For graduate students:

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

Leistungskontrollen / Prüfungen

- 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.

Lehrveranstaltungen / SWS

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

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

Arbeitsaufwand

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

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

Sprache English

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, 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.

Inhalt

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

Literaturhinweise

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

Digital Transmission & Signal Processing DTSP

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	4	at least every two years	1 semester	6	9

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

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

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

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

Arbeitsaufwand

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

Modulnote Final exam mark

Sprache English

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.

Literaturhinweise

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

Weitere Informationen

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

Distributed Systems DS

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	4	at least every two years	1 semester	6	9

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

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

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

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

Arbeitsaufwand

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

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

Sprache English

Lernziele / Kompetenzen

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

Inhalt

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

Literaturhinweise

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

Geometric Modelling GM

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Hans-Peter Seidel

Dozent/inn/en Prof. Dr. Hans-Peter Seidel
Dr. Rhaleb Zayer

Zulassungsvoraussetzungen calculus and basic programming skills

Leistungskontrollen / Prüfungen

- 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.

Lehrveranstaltungen / SWS

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

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

Arbeitsaufwand

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

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

Sprache English

Lernziele / Kompetenzen

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).

Inhalt

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

Literaturhinweise

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

Image Processing and Computer Vision — IPCV

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Joachim Weickert

Dozent/inn/en Prof. Dr. Joachim Weickert

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

Leistungskontrollen / Prüfungen

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

Lehrveranstaltungen / SWS

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

Arbeitsaufwand

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

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

Sprache English

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

```
Inhalt

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

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

Literaturhinweise

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

Information Retrieval and Data Mining — IRDM

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Gerhard Weikum

Dozent/inn/en Prof. Dr. Gerhard Weikum

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

Leistungskontrollen / Prüfungen

- 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)

Lehrveranstaltungen / SWS

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

Arbeitsaufwand

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

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

Sprache English

Lernziele / Kompetenzen

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

Inhalt

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

Literaturhinweise

Will be announced on the course web site.

Introduction to Computational Logic — ICL

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Gert Smolka

Dozent/inn/en Prof. Dr. Gert Smolka

Zulassungsvoraussetzungen keine

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

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

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

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

Sprache English

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

Literaturhinweise

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

Optimization Opti

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	4	at least every two years	1 semester	6	9

Modulverantwortliche/r Prof. Dr. Kurt Mehlhorn

Dozent/inn/en Prof. Dr. Kurt Mehlhorn
Dr. Andreas Karrenbauer

Zulassungsvoraussetzungen For graduate students: none

Leistungskontrollen / Prüfungen

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

Lehrveranstaltungen / SWS

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

Arbeitsaufwand

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

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

Sprache English

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

Literaturhinweise

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

Semantics

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1-3	**4**	**at least every two years**	**1 semester**	**6**	**9**

Modulverantwortliche/r Prof. Dr. Gert Smolka

Dozent/inn/en Prof. Dr. Gert Smolka

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

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

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

Sprache English

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

Literaturhinweise

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

Modulbereich 4

Ringvorlesungen

Ringvorlesung Cybersecurity

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1	**4**	**jedes Wintersemester**	**1 Semester**	**3**	**5**

Modulverantwortliche/r Prof. Dr. Andreas Zeller

Dozent/inn/en Professor/inn/en und Forscher/inn/en der Cybersicherheit

Zulassungsvoraussetzungen keine

Leistungskontrollen / Prüfungen Regelmäßige Teilnahme an der Ringvorlesung; Studierende verfassen zu fünf der vorgestellten Themen eine Analyse möglicher Umsetzungen in marktfähige Produkte, deren potenzielle Kunden und deren Marktchancen und -Risiken. Gruppenarbeiten von bis zu drei Teilnehmern sind möglich, sofern die Einzelbeiträge gekennzeichnet sind.

Lehrveranstaltungen / SWS `2 SWS Vorlesung`

Arbeitsaufwand

```
  30 h Präsenzstudium
+ 120 h Eigenstudium
= 150 h (= 5 ECTS)
```

Modulnote Wird bestimmt aus Qualität der schriftlichen Ausarbeitungen. Details werden vom Dozenten/von der Dozentin zu Beginn der Vorlesung bekanntgegeben.

Sprache Englisch

Lernziele / Kompetenzen

- Überblick über die zentralen wissenschaftlichen Fragestellungen der Cybersecurity und Informatik
- Einblick in Kompetenzen des Saarbrücker Informatik-Campus Ideenbildung

Inhalt

Querschnitt durch die Forschungsthemen der Saarbrücker Cybersecurity und Informatik. Die Themen spannen einen attraktiven Bogen von aktuellster Forschung zu anspruchsvollen Problemen der industriellen Praxis.

Literaturhinweise

Material wird passend zu den zu den einzelnen Vorträgen bereitgestellt.

Modulbereich 5

Seminare

Seminar

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
1	4	every semester	1 semester	2	7

Modulverantwortliche/r Dean of Studies of the Faculty of Mathematics and Computer Science
Dean of Studies of the Department of Computer Science

Dozent/inn/en Lecturers of the department

Zulassungsvoraussetzungen Basic knowledge of the relevant sub-field of the study program.

Leistungskontrollen / Prüfungen
- Thematic presentation with subsequent discussion
- Active participation in the discussion
- short written report and/or project possible

Lehrveranstaltungen / SWS `2 h seminar (weekly)`

Arbeitsaufwand
```
  30 h of lectures and exercises
+ 180 h project work
= 210 h (= 7 ECTS)
```

Modulnote Will be determined from the performance in the presentation and the written report and/or the seminar project. The exact modalities will be announced by the respective instructor.

Sprache English or German

Lernziele / Kompetenzen

At the end of the seminar, students have primarily gained a deep understanding of current or fundamental aspects of a specific subfield of computer science.

They have gained further competence in independent scientific research, classifying, summarizing, discussing, criticizing and presenting scientific findings.

Inhalt

Largely independent research of the seminar topic:

- Reading and understanding of scientific papers
- Analysis and evaluation of scientific papers
- Discussion of the scientific work in the group
- Analyzing, summarizing and reporting the specific topic
- Developing common standards for scientific work
- Presentation techniques

Specific in-depth study related to the individual topic of the seminar.

The typical procedure of a seminar is usually as follows:

- Preparatory discussions for topic selection
- Regular meetings with discussion of selected presentations
- if applicable, work on a project related to the topic
- Presentation and, if necessary, writing a report on one of the presentations

Literaturhinweise

Material is selected according to the topic.

Weitere Informationen

The seminars available will be announced prior to the beginning of the semester and will vary by study programme.

Modulbereich 6

Master-Seminar und -Arbeit

Master Seminar

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
3	4	every semester	1 semester	2	12

Modulverantwortliche/r Dean of Studies of the Faculty of Mathematics and Computer Science
Study representative of computer science

Dozent/inn/en Professors of the department

Zulassungsvoraussetzungen Acquisition of at least 30 CP

Leistungskontrollen / Prüfungen

- Preparation of the relevant scientific literature
- Written elaboration of the topic of the master thesis
- Presentation about the planned topic with subsequent discussion
- Active participation in the discussion

Lehrveranstaltungen / SWS `2 h seminar (weekly)`

Arbeitsaufwand

```
  30 h seminar
+  40 h contact with supervisor
+ 290 h private study
= 360 h (= 12 ECTS)
```

Modulnote graded

Sprache English or German

Lernziele / Kompetenzen

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.

Inhalt

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

Literaturhinweise

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

Master Thesis

Studiensem.	Regelst.sem.	Turnus	Dauer	SWS	ECTS
4	**4**	**every semester**	**6 months**	**-**	**30**

Modulverantwortliche/r Dean of Studies of the Faculty of Mathematics and Computer Science
Study representative of computer science

Dozent/inn/en Professors of the department

Zulassungsvoraussetzungen Successful completion of the *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 none

Arbeitsaufwand

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

Modulnote Grading of the Master Thesis

Sprache English or German

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.

Literaturhinweise

According to the topic