TUHH
Hamburg
University of
Technology

Module Manual

Master of Science (M.Sc.)

Computer Science

Cohort: Winter Term 2022

Updated: 8th May 2025

Table of Contents

Program description

Content

Core Qualification

Module M0523: Business & Management

Module Responsible	Prof. Matthias Meyer
Admission Requirements	None
Recommended Previous Knowledge	None
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students are able to find their way around selected special areas of management within the scope of business management. • Students are able to explain basic theories, categories, and models in selected special areas of business management. • Students are able to interrelate technical and management knowledge.
Skills	• Students are able to apply basic methods in selected areas of business management. • Students are able to explain and give reasons for decision proposals on practical issues in areas of business management.
Personal Competence	
Social Competence	• Students are able to communicate in small interdisciplinary groups and to jointly develop solutions for complex problems
Autonomy	• Students are capable of acquiring necessary knowledge independently by means of research and preparation of material.
Workload in Hours	Depends on choice of courses
Credit points	6

Courses
Information regarding lectures and courses can be found in the corresponding module handbook published separately.

Module M0524: Non-technical Courses for Master

Module Responsible	Dagmar Richter
Admission Requirements	None
Recommended Previous Knowledge	None
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	**The Nontechnical Academic Programms (NTA)** imparts skills that, in view of the TUHH's training profile, professional engineering studies require but are not able to cover fully. Self-reliance, self-management, collaboration and professional and personnel management competences. The department implements these training objectives in its **teaching architecture**, in its **teaching and learning arrangements**, in **teaching areas** and by means of teaching offerings in which students can qualify by opting for **specific competences** and a **competence level** at the Bachelor's or Master's level. The teaching offerings are pooled in two different catalogues for nontechnical complementary courses. **The Learning Architecture** consists of a cross-disciplinarily study offering. The centrally designed teaching offering ensures that courses in the nontechnical academic programms follow the specific profiling of TUHH degree courses. The learning architecture demands and trains independent educational planning as regards the individual development of competences. It also provides orientation knowledge in the form of "profiles". The subjects that can be studied in parallel throughout the student's entire study program - if need be, it can be studied in one to two semesters. In view of the adaptation problems that individuals commonly face in their first semesters after making the transition from school to university and in order to encourage individually planned semesters abroad, there is no obligation to study these subjects in one or two specific semesters during the course of studies. **Teaching and Learning Arrangements** provide for students, separated into B.Sc. and M.Sc., to learn with and from each other across semesters. The challenge of dealing with interdisciplinarity and a variety of stages of learning in courses are part of the learning architecture and are deliberately encouraged in specific courses. **Fields of Teaching** are based on research findings from the academic disciplines cultural studies, social studies, arts, historical studies, communication studies, migration studies and sustainability research, and from engineering didactics. In addition, from the winter semester 2014/15 students on all Bachelor's courses will have the opportunity to learn about business management and start-ups in a goal-oriented way. The fields of teaching are augmented by soft skills offers and a foreign language offer. Here, the focus is on encouraging goal-oriented communication skills, e.g. the skills required by outgoing engineers in international and intercultural situations. **The Competence Level** of the courses offered in this area is different as regards the basic training objective in the Bachelor's and Master's fields. These differences are reflected in the practical examples used, in content topics that refer to different professional application contexts, and in the higher scientific and theoretical level of abstraction in the B.Sc. This is also reflected in the different quality of soft skills, which relate to the different team positions and different group leadership functions of Bachelor's and Master's graduates in their future working life. **Specialized Competence (Knowledge)** Students can • explain specialized areas in context of the relevant non-technical disciplines, • outline basic theories, categories, terminology, models, concepts or artistic techniques in the disciplines represented in the learning area, • different specialist disciplines relate to their own discipline and differentiate it as well as make connections, • sketch the basic outlines of how scientific disciplines, paradigms, models, instruments, methods and forms of representation in the specialized sciences are subject to individual and socio-cultural interpretation and historicity, • Can communicate in a foreign language in a manner appropriate to the subject.
Skills	**Professional Competence (Skills)** In selected sub-areas students can • apply basic and specific methods of the said scientific disciplines, • aquestion a specific technical phenomena, models, theories from the viewpoint of another, aforementioned specialist discipline, • to handle simple and advanced questions in aforementioned scientific disciplines in a sucsessful manner, • justify their decisions on forms of organization and application in practical questions in contexts that go beyond the technical relationship to the subject.
Personal Competence	
Social Competence	**Personal Competences (Social Skills)**

	Students will be able • to learn to collaborate in different manner, • to present and analyze problems in the abovementioned fields in a partner or group situation in a manner appropriate to the addressees, • to express themselves competently, in a culturally appropriate and gender-sensitive manner in the language of the country (as far as this study-focus would be chosen), • to explain nontechnical items to auditorium with technical background knowledge.
Autonomy	**Personal Competences (Self-reliance)** Students are able in selected areas • to reflect on their own profession and professionalism in the context of real-life fields of application • to organize themselves and their own learning processes • to reflect and decide questions in front of a broad education background • to communicate a nontechnical item in a competent way in writen form or verbaly • to organize themselves as an entrepreneurial subject country (as far as this study-focus would be chosen)
Workload in Hours	Depends on choice of courses
Credit points	6

Courses
Information regarding lectures and courses can be found in the corresponding module handbook published separately.

Module M1563: Research Project Computer Science

Courses

Title	Typ	Hrs/wk	CP
Research Project Computer Science (L2353)	Projection Course	8	12

Module Responsible	Dozenten des SD E
Admission Requirements	None
Recommended Previous Knowledge	Basic knowledge and techniques from the Master courses in the semesters 1 and 2.
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students are able to acquire advanced knowledge in a subfield of Computer Science and can independently acquire deeper knowledge in the field.
Skills	The students are able to formulate the scientific problems to be considered and to work out solutions in an independent manner and to realize them.
Personal Competence	
Social Competence	The students are able to discuss proposals for solutions of scientific problems within the team. They are able to present the results in a clear and well structured manner.
Autonomy	The students can provide a scientific work in a timely manner and document the results in a detailed and well readable form. They are able to actively follow anticipate the presentations of other students such that eventually a scientific discussion comes up.
Workload in Hours	Independent Study Time 248, Study Time in Lecture 112
Credit points	12
Course achievement	None
Examination	Study work
Examination duration and scale	Vortrag
Assignment for the Following Curricula	Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory

Course L2353: Research Project Computer Science

Typ	Projection Course
Hrs/wk	8
CP	12
Workload in Hours	Independent Study Time 248, Study Time in Lecture 112
Lecturer	Dozenten des SD E
Language	DE/EN
Cycle	WiSe
Content	Current research topics of the chosen areas of specialization
Literature	Wird vom Veranstalter bekanntgegeben.

Specialization I. Computer and Software Engineering

Module M0753: Software Verification

Courses			
Title	**Typ**	**Hrs/wk**	**CP**
Software Verification (L0629)	Lecture	2	3
Software Verification (L0630)	Recitation Section (small)	2	3

Module Responsible	Prof. Sibylle Schupp
Admission Requirements	None
Recommended Previous Knowledge	• Automata theory and formal languages • Computational logic • Object-oriented programming, algorithms, and data structures • Functional programming or procedural programming • Concurrency
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students apply the major verification techniques in model checking and deductive verification. They explain in formal terms syntax and semantics of the underlying logics, and assess the expressivity of different logics as well as their limitations. They classify formal properties of software systems. They find flaws in formal arguments, arising from modeling artifacts or underspecification.
Skills	Students formulate provable properties of a software system in a formal language. They develop logic-based models that properly abstract from the software under verification and, where necessary, adapt model or property. They construct proofs and property checks by hand or using tools for model checking or deductive verification, and reflect on the scope of the results. Presented with a verification problem in natural language, they select the appropriate verification technique and justify their choice.
Personal Competence	
Social Competence	Students discuss relevant topics in class. They defend their solutions orally. They communicate in English.
Autonomy	Using accompanying on-line material for self study, students can assess their level of knowledge continuously and adjust it appropriately. Working on exercise problems, they receive additional feedback. Within limits, they can set their own learning goals. Upon successful completion, students can identify and precisely formulate new problems in academic or applied research in the field of software verification. Within this field, they can conduct independent studies to acquire the necessary competencies and compile their findings in academic reports. They can devise plans to arrive at new solutions or assess existing ones.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory** Yes · **Bonus** 15 % · **Form** Excercises · **Description**
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems: Compulsory Information and Communication Systems: Specialisation Communication Systems, Focus Software: Elective Compulsory International Management and Engineering: Specialisation II. Information Technology: Elective Compulsory

Course L0629: Software Verification

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	WiSe
Content	• ◦ Model checking (bounded model checking, CTL, LTL) ◦ Real-time model checking (TCTL, timed automata) ◦ Deductive verification (Hoare logic) ◦ Tool support ◦ Recent developments of verification techniques and applications
Literature	• C. Baier and J-P. Katoen, Principles of Model Checking, MIT Press 2007. • M. Huth and M. Bryan, Logic in Computer Science. Modelling and Reasoning about Systems, 2nd Edition, 2004. • Selected Research Papers

Course L0630: Software Verification	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M0942: Software Security

Courses

Title	Typ	Hrs/wk	CP
Software Security (L1103)	Lecture	2	3
Software Security (L1104)	Recitation Section (small)	2	3

Module Responsible	Prof. Riccardo Scandariato
Admission Requirements	None
Recommended Previous Knowledge	Familiarity with C/C++, web programming
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students can • name the main causes for security vulnerabilities in software • explain current methods for identifying and avoiding security vulnerabilities • explain the fundamental concepts of code-based access control
Skills	Students are capable of • performing a software vulnerability analysis • developing secure code
Personal Competence	
Social Competence	None
Autonomy	Students are capable of acquiring knowledge independently from professional publications, technical standards, and other sources, and are capable of applying newly acquired knowledge to new problems.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	120 minutes
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems: Elective Compulsory

Course L1103: Software Security

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Riccardo Scandariato
Language	EN
Cycle	WiSe
Content	• Reliabilty and Software Security • Attacks exploiting character and integer representations • Buffer overruns • Vulnerabilities in memory managemet: double free attacks • Race conditions • SQL injection • Cross-site scripting and cross-site request forgery • Testing for security; taint analysis • Type safe languages • Development proceses for secure software • Code-based access control
Literature	M. Howard, D. LeBlanc: Writing Secure Code, 2nd edition, Microsoft Press (2002) G. Hoglund, G. McGraw: Exploiting Software, Addison-Wesley (2004) L. Gong, G. Ellison, M. Dageforde: Inside Java 2 Platform Security, 2nd edition, Addison-Wesley (2003) B. LaMacchia, S. Lange, M. Lyons, R. Martin, K. T. Price: .NET Framework Security, Addison-Wesley Professional (2002) D. Gollmann: Computer Security, 3rd edition (2011)

Course L1104: Software Security	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Riccardo Scandariato
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1694: Security of Cyber-Physical Systems

Courses

Title	Typ	Hrs/wk	CP
Security of Cyber-Physical Systems (L2691)	Lecture	2	3
Security of Cyber-Physical Systems (L2692)	Recitation Section (small)	2	3

Module Responsible	Prof. Sibylle Fröschle
Admission Requirements	None
Recommended Previous Knowledge	IT security, programming skills, statistics
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	The students know and can explain - the threats posed by cyber attacks to cyber-physical systems (CPS) - concrete attacks at a technical level, e.g. on bus systems - security solutions specific to CPS with their capabilities and limitations - examples of security architectures for CPS and the requirements they guarantee - standard security engineering processes for CPS
Skills	The students are able to - identify security threats and assess the risks for a given CPS - apply attack toolkits to analyse a networked control system, and detect attacks beyond those taught in class - identify and apply security solutions suitable to the requirements - follow security engineering processes to develop a security architecture for a given CPS - recognize challenges and limitations, e.g. posed by novel types of attack
Personal Competence	
Social Competence	The students are able to - expertly discuss security risks and incidents of CPS and their mitigation in a solution-oriented fashion with experts and non-experts - foster a security culture with respect to CPS and the corresponding critical infrastructures
Autonomy	The students are able to - follow up and critically assess current developments in the security of CPS including relevant security incidents - master a new topic within the area by self-study and self-initiated interaction with experts and peers.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory**: No; **Bonus**: 10 %; **Form**: Excercises; **Description**: Die Übungsaufgaben finden semesterbegleitend statt.
Examination	Written exam
Examination duration and scale	120 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems, Focus Software and Signal Processing: Elective Compulsory

Course L2691: Security of Cyber-Physical Systems	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sibylle Fröschle
Language	EN
Cycle	WiSe
Content	Embedded systems in energy, production, and transportation are currently undergoing a technological transition to highly networked automated cyber-physical systems (CPS). Such systems are potentially vulnerable to cyber attacks, and these can have physical impact. In this course we investigate security threats, solutions and architectures that are specific to CPS. The topics are as follows: Fundamentals and motivating examples Networked and embedded control systems Bus system level attacks Intruder detection systems (IDS), in particular physics-based IDS System security architectures, including cryptographic solutions Adversarial machine learning attacks in the physical world Aspects of Location and Localization Wireless networks and infrastructures for critical applications Communication security architectures and remaining threats Intruder detection systems (IDS), in particular data-centric IDS Resilience against multi-instance attacks Security Engineering of CPS: Process and Norms
Literature	Recent scientific papers and reports in the public domain.

Course L2692: Security of Cyber-Physical Systems	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sibylle Fröschle
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M0926: Distributed Algorithms

Courses			
Title	**Typ**	**Hrs/wk**	**CP**
Distributed Algorithms (L1071)	Lecture	2	3
Distributed Algorithms (L1072)	Recitation Section (large)	2	3

Module Responsible	Prof. Volker Turau
Admission Requirements	None
Recommended Previous Knowledge	• Algorithms and data structures • Distributed systems • Discrete mathematics • Graph theory
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students know the main abstractions of distributed algorithms (synchronous/asynchronous model, message passing and shared memory model). They are able to describe complexity measures for distributed algorithms (round , message and memory complexity). They explain well known distributed algorithms for important problems such as leader election, mutual exclusion, graph coloring, spanning trees. They know the fundamental techniques used for randomized algorithms.
Skills	Students design their own distributed algorithms and analyze their complexity. They make use of known standard algorithms. They compute the complexity of randomized algorithms.
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	45 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory

Course L1071: Distributed Algorithms

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Volker Turau
Language	DE/EN
Cycle	WiSe
Content	• Leader Election • Colorings & Independent Sets • Tree Algorithms • Minimal Spanning Trees • Randomized Distributed Algorithms • Mutual Exclusion
Literature	1. David Peleg: Distributed Computing - A Locality-Sensitive Approach. SIAM Monograph, 2000 2. Gerard Tel: Introduction to Distributed Algorithms, Cambridge University Press, 2nd edition, 2000 3. Nancy Lynch: Distributed Algorithms. Morgan Kaufmann, 1996 4. Volker Turau: Algorithmische Graphentheorie. Oldenbourg Wissenschaftsverlag, 3. Auflage, 2004.

Course L1072: Distributed Algorithms

Typ	Recitation Section (large)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Volker Turau
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1749: Energy Efficiency in Embedded Systems

Courses

Title	Typ	Hrs/wk	CP
Energy Efficiency in Embedded Systems (L2870)	Lecture	2	3
Energy Efficiency in Embedded Systems (L2872)	Project-/problem-based Learning	2	2
Energy Efficiency in Embedded Systems (L2871)	Recitation Section (large)	1	1

Module Responsible	Prof. Ulf Kulau
Admission Requirements	None
Recommended Previous Knowledge	• Computer Engineering (mandatory) • Programming Skills in C (mandatory) • Computer Architecture (recommended)
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	**Motivation:** In the field of computer science we have only limited possibilities to influence the efficiency of the hardware directly, respectively we are dependent on the manufacturers (e.g. of microcontrollers). However, in order to exploit the full potential of the hardware we are given at the system level, we need a deeper understanding of the background, processes and mechanisms of power dissipation in embedded systems. Where does the power dissipation come from, what happens at the hardware level, what mechanisms can I use directly/indirectly, what is the tradeoff between flexibility and efficiency,.... are only a few questions, which will be elaborated and discussed in this event. **Contents of teaching:** • Motivation and power dissipation on semiconductor level • Power dissipation of digital circuits, inparticular CMOS • Power Management in Hard- and Software (Sleep Modes, DVS, FS, Undervolting) • Energy efficient system design (applications) • Energy Harvesting and Transiently Powered Computing (TPC)
Skills	Upon completion of this module, students will have a deeper understanding of hardware and software mechanisms for evaluating and developing energy-efficient embedded systems • They have a deeper understanding of the electrotechnical basics of power dissipation in digital systems • They can analyze the power dissipation of systems at any level and apply appropriate methods to increase efficiency • They can use a variety of standard techniques to achieve "Energy Efficiency by Design" • They can model, evaluate as well as implement energy-autonomous systems
Personal Competence	
Social Competence	As part of the module, concepts learned in the lecture will be implemented on a hardware platform within small groups. Students learn to work in a team and to develop solutions together. Specific tasks are worked on within the group, whereby cross-group collaboration (exchange) also takes place. The second part is a challenge-based project in which the groups find the most energy-efficient solutions possible in healthy competition with each other. This strengthens the cohesion in the groups and reinforces mutual motivation, support and creativity.
Autonomy	After completing this module, students will be able to independently develop, optimize and evaluate solutions for embedded systems based on the knowledge they have acquired and further technical literature.
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	25 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Electrical Engineering: Specialisation Nanoelectronics and Microsystems Technology: Elective Compulsory Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory

Course L2870: Energy Efficiency in Embedded Systems	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Ulf Kulau
Language	DE/EN
Cycle	WiSe
Content	**Motivation:** In the field of computer science we have only limited possibilities to influence the efficiency of the hardware directly, respectively we are dependent on the manufacturers (e.g. of microcontrollers). However, in order to exploit the full potential of the hardware we are given at the system level, we need a deeper understanding of the background, processes and mechanisms of power dissipation in embedded systems. Where does the power dissipation come from, what happens at the hardware level, what mechanisms can I use directly/indirectly, what is the tradeoff between flexibility and efficiency,.... are only a few questions, which will be elaborated and discussed in this event. **Contents of teaching:** • Motivation and power dissipation on semiconductor level • Power dissipation of digital circuits, inparticular CMOS • Power Management in Hard- and Software (Sleep Modes, DVS, FS, Undervolting) • Energy efficient system design (applications) • Energy Harvesting and Transiently Powered Computing (TPC)
Literature	DE: Die Vorlesung basiert af einer Vielzahl von Quellen, welche in [1.] angegeben sind. ENG: The lecture is based on multiple sources which are listed in [1.]. 1. Kulau, Ulf: **Course: Energy Efficiency in Embedded Systems-A System-Level Perspective for Computer Scientists**, EWME, 2018. 2. Harris, David, and N. Weste: **CMOS VLSI Design** ed., Pearson Education, 2010 3. Rabaey, Jan: **Low Power Design Essentials** (Integrated Circuits and Systems), Springer, 2009

Course L2872: Energy Efficiency in Embedded Systems	
Typ	Project-/problem-based Learning
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Ulf Kulau
Language	DE/EN
Cycle	WiSe
Content	In this project-based exercise, the learned aspects for achieving energy-efficient embedded systems are implemented and consolidated in practical environments in a small project. First, a tool set for the implementation of energy efficiency mechanisms is implemented in common exercises by means of defined tasks. In the second part, a challenge-based exercise is carried out in which a system that is as efficient as possible is to be implemented independently. A system based on an AVR micro-controller is used, which can be operated autonomously by a Solar-Energy Harvester. 1. Task phase: 6 "hands-on" tasks to gain experience and to create a SW library. 2. Project phase: Implementation of an energy autonomous system with the goal of highest possible energy efficiency (Challenge)
Literature	

Course L2871: Energy Efficiency in Embedded Systems	
Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Ulf Kulau
Language	DE/EN
Cycle	WiSe
Content	In the lecture hall exercise, the theoertical basics taught in the lecture are deepened. This is done through in-depth discussion of relevant aspects, but also through calculation examples, in which a deeper understanding of the topic of energy efficiency in embedded systems is gained. Exercises will be distributed in advance and solutions will be presented in the lecture hall exercise. Contents of the exercise are as follows: • Basics and calculation of power dissipation on semiconductor • Power dissipation of CMOS using the example of an inverter • Influence of the activity factor and external components • DVS and scheduling • Evaluation to show the benefit of undervolting • Aspects of energy harvesting (MPPT) •
Literature	

Module M1400: Design of Dependable Systems

Courses

Title	Typ	Hrs/wk	CP
Designing Dependable Systems (L2000)	Lecture	2	3
Designing Dependable Systems (L2001)	Recitation Section (small)	2	3

Module Responsible	Prof. Görschwin Fey
Admission Requirements	None
Recommended Previous Knowledge	Basic knowledge about data structures and algorithms
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	In the following "dependable" summarizes the concepts Reliability, Availability, Maintainability, Safety and Security. Knowledge about approaches for designing dependable systems, e.g., • Structural solutions like modular redundancy • Algorithmic solutions like handling byzantine faults or checkpointing Knowledge about methods for the analysis of dependable systems
Skills	Ability to implement dependable systems using the above approaches. Ability to analyzs the dependability of systems using the above methods for analysis.
Personal Competence	
Social Competence	Students • discuss relevant topics in class and • present their solutions orally.
Autonomy	Using accompanying material students independently learn in-depth relations between concepts explained in the lecture and additional solution strategies.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory**: Yes; **Bonus**: None; **Form**: Subject theoretical and practical work; **Description**: Die Lösung einer Aufgabe ist Zulassungsvoraussetzung für die Prüfung. Die Aufgabe wird in Vorlesung und Übung definiert.
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems: Elective Compulsory Mechatronics: Specialisation System Design: Elective Compulsory Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory

Course L2000: Designing Dependable Systems	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Görschwin Fey
Language	DE/EN
Cycle	SoSe
Content	Description The term dependability comprises various aspects of a system. These are typically: • Reliability • Availability • Maintainability • Safety • Security This makes dependability a core aspect that has to be considered early in system design, no matter whether software, embedded systems or full scale cyber-physical systems are considered. Contents The module introduces the basic concepts for the design and the analysis of dependable systems. Design examples for getting practical hands-on-experience in dependable design techniques. The module focuses towards embedded systems. The following topics are covered: • Modelling • Fault Tolerance • Design Concepts • Analysis Techniques
Literature	

Course L2001: Designing Dependable Systems	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Görschwin Fey
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M1397: Model Checking - Proof Engines and Algorithms

Courses

Title	Typ	Hrs/wk	CP
Model Checking - Proof Engines and Algorithms (L1979)	Lecture	2	3
Model Checking - Proof Engines and Algorithms (L1980)	Recitation Section (small)	2	3

Module Responsible	Prof. Görschwin Fey
Admission Requirements	None
Recommended Previous Knowledge	Basic knowledge about data structures and algorithms
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students know • algorithms and data structures for model checking, • basics of Boolean reasoning engines and • the impact of specification and modelling on the computational effort for model checking.
Skills	Students can • explain and implement algorithms and data structures for model checking, • decide whether a given problem can be solved using Boolean reasoning or model checking, and • implement the respective algorithms.
Personal Competence	
Social Competence	Students • discuss relevant topics in class and • defend their solutions orally.
Autonomy	Using accompanying material students independently learn in-depth relations between concepts explained in the lecture and additional solution strategies.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory**: Yes; **Bonus**: None; **Form**: Subject theoretical and practical work; **Description**: Die Aufgabe wird im Rahmen von Volresung und Prüfung definiert. Die Lösung der Aufgabe ist Zulassungsvoraussetzung für die Prüfung.
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Information and Communication Systems: Specialisation Communication Systems, Focus Software: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems: Elective Compulsory

Course L1979: Model Checking - Proof Engines and Algorithms	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Görschwin Fey
Language	DE/EN
Cycle	SoSe

Content	Correctness is a major concern in embedded systems. Model checking can fully automatically proof formal properties about digital hardware or software. Such properties are given in temporal logic, e.g., to prove "No two orthogonal traffic lights will ever be green."
	
	And how do the underlying reasoning algorithms work so effectively in practice despite a computational complexity of NP hardness and beyond?
	
	But what are the limitations of model checking?
	How are the models generated from a given design?
	The lecture will answer these questions. Open source tools will be used to gather a practical experience.
	
	Among other topics, the lecture will consider the following topics:
	
	• Modelling digital Hardware, Software, and Cyber Physical Systems
	• Data structures, decision procedures and proof engines
	    ◦ Binary Decision Diagrams
	    ◦ And-Inverter-Graphs
	    ◦ Boolean Satisfiability
	    ◦ Satisfiability Modulo Theories
	• Specification Languages
	    ◦ CTL
	    ◦ LTL
	    ◦ System Verilog Assertions
	• Algorithms for
	    ◦ Reachability Analysis
	    ◦ Symbolic CTL Checking
	    ◦ Bounded LTL-Model Checking
	    ◦ Optimizations, e.g., induction, abstraction
	• Quality assurance

Literature	Edmund M. Clarke, Jr., Orna Grumberg, and Doron A. Peled. 1999. *Model Checking*. MIT Press, Cambridge, MA, USA. A. Biere, A. Biere, M. Heule, H. van Maaren, and T. Walsh. 2009. *Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications*. IOS Press, Amsterdam, The Netherlands, The Netherlands. Selected research papers

Course L1980: Model Checking - Proof Engines and Algorithms	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Görschwin Fey
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M1301: Software Testing

Courses

Title	Typ	Hrs/wk	CP
Software Testing (L1791)	Lecture	2	3
Software Testing (L1792)	Project-/problem-based Learning	2	3

Module Responsible	Prof. Sibylle Schupp
Admission Requirements	None
Recommended Previous Knowledge	• Software Engineering • Higher Programming Languages • Object-Oriented Programming • Algorithms and Data Structures • Experience with (Small) Software Projects • Statistics
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students explain the different phases of testing, describe fundamental techniques of different types of testing, and paraphrase the basic principles of the corresponding test process. They give examples of software development scenarios and the corresponding test type and technique. They explain algorithms used for particular testing techniques and describe possible advantages and limitations.
Skills	Students identify the appropriate testing type and technique for a given problem. They adapt and execute respective algorithms to execute a concrete test technique properly. They interpret testing results and execute corresponding steps for proper re-test scenarios. They write and analyze test specifications. They apply bug finding techniques for non-trivial problems.
Personal Competence	
Social Competence	Students discuss relevant topics in class. They defend their solutions orally. They communicate in English.
Autonomy	Students can assess their level of knowledge continuously and adjust it appropriately, based on feedback and on self-guided studies. Within limits, they can own learning goals. Upon successful completion, students can identify and precisely formulate new problems in academic or applied research in the field of testing. Within this field, they can conduct independent studies to acquire the necessary competencies and compile their findings in academic reports. T devise plans to arrive at new solutions or assess existing ones
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Subject theoretical and practical work
Examination duration and scale	Software
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Information and Communication Systems: Specialisation Communication Systems, Focus Software: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems, Focus Software and Signal Processing: Elective Compulsory

Course L1791: Software Testing	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	SoSe
Content	• Fundamentals of software testing • Model-based testing • Test automation • Criteria-based testing
Literature	• M. Pezze and M. Young, Software Testing and Analysis, John Wiley 2008. • P. Ammann and J. Offutt, "Introduction to Software Testing", 2nd edition 2016. • A. Zeller: "Why Programs Fail: A Guide to Systematic Debugging", 2nd edition 2012.

Course L1792: Software Testing	
Typ	Project-/problem-based Learning
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	SoSe
Content	• Fundamentals of software testing • Model-based testing • Test automation • Criteria-based testing
Literature	• M. Pezze and M. Young, Software Testing and Analysis, John Wiley 2008. • P. Ammann and J. Offutt, "Introduction to Software Testing", 2nd edition 2015.

Module M1427: Algorithmic Game Theory

Courses			
Title	**Typ**	**Hrs/wk**	**CP**
Algorithmic game theory (L2060)	Lecture	2	4
Algorithmic game theory (L2061)	Recitation Section (large)	2	2

Module Responsible	Prof. Matthias Mnich
Admission Requirements	None
Recommended Previous Knowledge	• Mathematics I • Mathematics II • Algorithms and Data Structures
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students can name the basic concepts in algorithmic game theory and mechanism design. They are able to explain them using appropriate examples. • Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples. • They know game and mechanism design strategies and can reproduce them.
Skills	• Students can model strategic interaction systems of agents with the help of the concepts studied in this course. Moreover, they are capable of analyzing their efficiency and equilibria, by applying established methods. • Students are able to discover and verify further logical connections between the concepts studied in the course. • For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.
Personal Competence	
Social Competence	• Students are able to work together in teams. They are capable to use mathematics as a common language. • In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.
Autonomy	• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. • Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory

Course L2060: Algorithmic game theory	
Typ	Lecture
Hrs/wk	2
CP	4
Workload in Hours	Independent Study Time 92, Study Time in Lecture 28
Lecturer	Prof. Matthias Mnich
Language	DE/EN
Cycle	SoSe
Content	Algorithmic game theory is a topic at the intersection of economics and computation. It deals with analyzing the behavior and interactions of strategic agents, who often try to maximize their incentives. The environment in which those agents interact is referred to as a game. We wish to understand if the agents can reach an "equilibrium", or steady state of the game, in which agents have no incentive to deviate from their chosen strategies. The algorithmic part is to design efficient methods to find equilibria in games, and to make recommendations to the agents so that they can quickly reach a state of personal satisfaction. We will also study mechanism design. In mechanism design, we wish to design markets and auctions and give strategic options to agents, so that they have an incentive to act rationally. We also wish to design the markets and auctions so that they are efficient, in the sense that all goods are cleared and agents do not overpay for the goods which they acquire. Topics: • basic equilibrium concepts (Nash equilibria, correlated equilibria, ...) • strategic actions (best-response dynamics, no-regret dynamics, ...) • auction design (revenue-maximizing auctions, Vickrey auctions) • stable matching theory (preference aggregations, kidney exchanges, ...) • price of anarchy and selfish routing (Braess' paradox, congestion games, ...)
Literature	• T. Roughgarden: Twenty Lectures on Algorithmic Game Theory, Cambridge University Press, 2016. • N. Nisan, T. Roughgarden, E. Tardos, V. Vazirani. Algorithmic Game Theory. Cambridge University Press, 2007.

Course L2061: Algorithmic game theory	
Typ	Recitation Section (large)
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Matthias Mnich
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M1248: Compilers for Embedded Systems

Courses

Title	Typ	Hrs/wk	CP
Compilers for Embedded Systems (L1692)	Lecture	3	4
Compilers for Embedded Systems (L1693)	Project-/problem-based Learning	1	2

Module Responsible	Prof. Heiko Falk
Admission Requirements	None
Recommended Previous Knowledge	Module "Embedded Systems" C/C++ Programming skills
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	The relevance of embedded systems increases from year to year. Within such systems, the amount of software to be executed on embedded processors grows continuously due to its lower costs and higher flexibility. Because of the particular application areas of embedded systems, highly optimized and application-specific processors are deployed. Such highly specialized processors impose high demands on compilers which have to generate code of highest quality. After the successful attendance of this course, the students are able • to illustrate the structure and organization of such compilers, • to distinguish and explain intermediate representations of various abstraction levels, and • to assess optimizations and their underlying problems in all compiler phases. The high demands on compilers for embedded systems make effective code optimizations mandatory. The students learn in particular, • which kinds of optimizations are applicable at the source code level, • how the translation from source code to assembly code is performed, • which kinds of optimizations are applicable at the assembly code level, • how register allocation is performed, and • how memory hierarchies can be exploited effectively. Since compilers for embedded systems often have to optimize for multiple objectives (e.g., average- or worst-case execution time, energy dissipation, code size), the students learn to evaluate the influence of optimizations on these different criteria.
Skills	After successful completion of the course, students shall be able to translate high-level program code into machine code. They will be enabled to assess which kind of code optimization should be applied most effectively at which abstraction level (e.g., source or assembly code) within a compiler. While attending the labs, the students will learn to implement a fully functional compiler including optimizations.
Personal Competence	
Social Competence	Students are able to solve similar problems alone or in a group and to present the results accordingly.
Autonomy	Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Electrical Engineering: Specialisation Information and Communication Systems: Elective Compulsory Aircraft Systems Engineering: Core Qualification: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Mechatronics: Specialisation System Design: Elective Compulsory Mechatronics: Technical Complementary Course: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory

Course L1692: Compilers for Embedded Systems	
Typ	Lecture
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Heiko Falk
Language	DE/EN
Cycle	SoSe
Content	• Introduction and Motivation • Compilers for Embedded Systems - Requirements and Dependencies • Internal Structure of Compilers • Pre-Pass Optimizations • HIR Optimizations and Transformations • Code Generation • LIR Optimizations and Transformations • Register Allocation • WCET-Aware Compilation • Outlook
Literature	• Peter Marwedel. Embedded System Design - Embedded Systems Foundations of Cyber-Physical Systems. 2 nd Edition, Springer, 2012. • Steven S. Muchnick. Advanced Compiler Design and Implementation. Morgan Kaufmann, 1997. • Andrew W. Appel. Modern compiler implementation in C. Oxford University Press, 1998.

Course L1693: Compilers for Embedded Systems	
Typ	Project-/problem-based Learning
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Heiko Falk
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M0556: Computer Graphics

Courses

Title	Typ	Hrs/wk	CP
Computer Graphics (L0145)	Lecture	2	3
Computer Graphics (L0768)	Recitation Section (small)	2	3

Module Responsible	Prof. Tobias Knopp
Admission Requirements	None
Recommended Previous Knowledge	• Linear Algebra (in particular matrix/vector computation) • Basic programming skills in C/C++
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students can explain and describe basic algorithms in 3D computer graphics.
Skills	Students are capable of • implementing a basic 3D rendering pipeline. This consists of projecting simple 3D structures (e.g. cube, spheres) onto a 2D surface using a virtual camera. • apply geometric transformations (e.g. rotation, scaling) in 2D and 3D computer graphics. • using well-known 2D/3D APIs (OpenGL, Cairo) for solving a given problem statement.
Personal Competence	
Social Competence	Students can collaborate in a small team on the realization and validation of a 3D computer graphics pipeline.
Autonomy	• Students are able to solve simple tasks independently with reference to the contents of the lectures and the exercise sets. • Students are able to solve detailed problems independently with the aid of the tutorial’s programming task.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems, Focus Software and Signal Processing: Elective Compulsory Information and Communication Systems: Specialisation Communication Systems, Focus Signal Processing: Elective Compulsory International Management and Engineering: Specialisation II. Information Technology: Elective Compulsory

Course L0145: Computer Graphics	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Tobias Knopp
Language	EN
Cycle	SoSe
Content	Computer graphics and animation are leading to an unprecedented visual revolution. The course deals with its technological foundations: • Object-oriented Computer Graphics • Projections and Transformations • Polygonal and Parametric Modelling • Illuminating, Shading, Rendering • Computer Animation Techniques • Kinematics and Dynamics Effects Students will be be working on a series of mini-projects which will eventually evolve into a final project. Learning computer graphics and animation resembles learning a musical instrument. Therefore, doing your projects well and in time is essential for performing well on this course.
Literature	Alan H. Watt: 3D Computer Graphics. Harlow: Pearson (3rd ed., repr., 2009). Dariush Derakhshani: Introducing Autodesk Maya 2014. New York, NY : Wiley (2013).

Course L0768: Computer Graphics	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Tobias Knopp
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M1685: Selected Aspects in Computer Science

Courses

Title	Typ	Hrs/wk	CP
Selected Aspects in Computer Science (L2672)	Lecture	3	4
Selected Aspects in Computer Science (L2673)	Recitation Section (small)	1	2

Module Responsible	Prof. Görschwin Fey
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory

Course L2672: Selected Aspects in Computer Science

Typ	Lecture
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Dozenten des SD E
Language	DE/EN
Cycle	WiSe/SoSe
Content	
Literature	

Course L2673: Selected Aspects in Computer Science

Typ	Recitation Section (small)
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Dozenten des SD E
Language	DE/EN
Cycle	WiSe/SoSe
Content	See interlocking course
Literature	See interlocking course

Module M1741: Operating System Construction

Courses			
Title	**Typ**	**Hrs/wk**	**CP**
Operating System Construction (L2812)	Lecture	2	3
Operating System Construction (L2814)	Project-/problem-based Learning	3	2
Operating System Construction (L2813)	Recitation Section (large)	1	1

Module Responsible	Prof. Christian Dietrich
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 96, Study Time in Lecture 84
Credit points	6
Course achievement	**Compulsory** No; **Bonus** 20 %; **Form** Subject theoretical and practical work; **Description**
Examination	Oral exam
Examination duration and scale	25 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory

Course L2812: Operating System Construction	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Christian Dietrich
Language	DE
Cycle	SoSe
Content	
Literature	

Course L2814: Operating System Construction	
Typ	Project-/problem-based Learning
Hrs/wk	3
CP	2
Workload in Hours	Independent Study Time 18, Study Time in Lecture 42
Lecturer	Prof. Christian Dietrich
Language	DE
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Course L2813: Operating System Construction	
Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Christian Dietrich
Language	DE
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M1682: Secure Software Engineering

Courses

Title	Typ	Hrs/wk	CP
Secure Software Engineering (L2667)	Lecture	2	3
Secure Software Engineering (L2668)	Project-/problem-based Learning	2	3

Module Responsible	Prof. Riccardo Scandariato
Admission Requirements	None
Recommended Previous Knowledge	Familiarity with basic software engineering concepts (e.g., requirements, design) and basic security concepts (e.g., confidentiality, integrity, availability)
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students can: • Elicit security requirements in a software project • Model and document security measures in a software design • Use threat and risk analysis techniques • Understand how security code reviews are performed • Understand the core definitions of concepts related to privacy • Understand privacy enhancing technologies
Skills	Select appropriate security assurance techniques to be used in a security assurance program
Personal Competence	
Social Competence	None
Autonomy	Students can apply the knowledge acquired throughout the course to the resolution of industrial case studies. Students should also be capable to acquire new knowledge independently from academic publications, techical standards, and white papers.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	120 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Information and Communication Systems: Specialisation Communication Systems, Focus Software: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems, Focus Software and Signal Processing: Elective Compulsory

Course L2667: Secure Software Engineering

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Riccardo Scandariato
Language	EN
Cycle	SoSe
Content	• Secure software development processes and maturity models • Techniques to define security requirements • Techniques to create, document and analyse the design of secure applications • Threat and risk analysis techniques • Security code reviews • Program repair techniques for security vulnerabilities • Privacy engineering
Literature	Sindre, G. and Opdahl, A.L., 2005. Eliciting security requirements with misuse cases. Requirements engineering, 10(1), pp.34-44. Fontaine, P.J., Van Lamsweerde, A., Letier, E. and Darimont, R., 2001. Goal-oriented elaboration of security requirements. Mead, N.R. and Stehney, T., 2005. Security quality requirements engineering (SQUARE) methodology. ACM SIGSOFT Software Engineering Notes, 30(4), pp.1-7. Mirakhorli, M., Shin, Y., Cleland-Huang, J. and Cinar, M., 2012, June. A tactic-centric approach for automating traceability of quality concerns. In 2012 34th international conference on software engineering (ICSE) (pp. 639-649). IEEE. Jürjens, J., UMLsec: Extending UML for secure systems development, International Conference on The Unified Modeling Language, 2002 Lund, M.S., Solhaug, B. and Stølen, K., 2011. A guided tour of the CORAS method. In Model-Driven Risk Analysis (pp. 23-43). Springer, Berlin, Heidelberg. Howard, M.A., 2006. A process for performing security code reviews. IEEE Security & privacy, 4(4), pp.74-79 Diaz, C. and Gürses, S., 2012. Understanding the landscape of privacy technologies. Proceedings of the information security summit, 12, pp.58-63.

Course L2668: Secure Software Engineering	
Typ	Project-/problem-based Learning
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Riccardo Scandariato
Language	EN
Cycle	SoSe
Content	• Secure software development processes and maturity models • Techniques to define security requirements • Techniques to create, document and analyse the design of secure applications • Threat and risk analysis techniques • Security code reviews • Program repair techniques for security vulnerabilities • Privacy engineering
Literature	

Module M1774: Advanced Internet Computing

Courses

Title	Typ	Hrs/wk	CP
Advanced Internet Computing (L2916)	Lecture	2	3
Advanced Internet Computing (L2917)	Project-/problem-based Learning	2	3

Module Responsible	Prof. Stefan Schulte
Admission Requirements	None
Recommended Previous Knowledge	Good programming skills are necessary. Previous knowledge in the field of distributed systems is helpful.
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	After successful completion of the course, students are able to: • Describe basic concepts of Cloud Computing, the Internet of Things (IoT), and blockchain technologies • Discuss and assess critical aspects of Cloud Computing, the IoT, and blockchain technologies • Select and apply cloud and IoT technologies for particular application areas • Design and develop practical solutions for the integration of smart objects in IoT, Cloud, and blockchain software • Implement IoT services
Skills	The students acquire the ability to model Internet-based distributed systems and to work with these systems. This comprises especially the ability to select and utilize fitting technologies for different application areas. Furthermore, students are able to critically assess the chosen technologies.
Personal Competence	
Social Competence	Students can work on complex problems both independently and in teams. They can exchange ideas with each other and use their individual strengths to solve the problem.
Autonomy	Students are able to independently investigate a complex problem and assess which competencies are required to solve it.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Subject theoretical and practical work
Examination duration and scale	Group project incl. presentation (50 %), written exam (60 min, 50 %)
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Information and Communication Systems: Specialisation Communication Systems, Focus Software: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems, Focus Networks: Elective Compulsory

Course L2916: Advanced Internet Computing

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Stefan Schulte
Language	EN
Cycle	SoSe
Content	This lecture discusses modern Internet-based distributed systems in three blocks: (i) Cloud computing, (ii) the Internet of Things, and (iii) blockchain technologies. The following topics will be covered in the single lectures: • Cloud Computing • Elastic Computing • Technologies for identification for the IoT: RFID & EPC • Communication in the IoT: Standards and protocols • Security and trust in the IoT: Concerns and solution approaches • Edge and Fog Computing • Application areas: Smart factories, smart cities, smart healthcare • Blockchain technologies • Consensus
Literature	Will be discussed in the lecture

Course L2917: Advanced Internet Computing	
Typ	Project-/problem-based Learning
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Stefan Schulte
Language	EN
Cycle	SoSe
Content	This project-/problemoriented part of the module augments the theoretical content of the lecture by a concrete technical problem, which needs to be solved by the students in group work during the semester. Possible topics are (blockchain-based) sensor data integration, Big Data processing, Cloud-based redundant data storages, and Cloud-based Onion Routing.
Literature	Will be discussed in the lecture.

Module M1773: Cybersecurity Data Science

Courses

Title	Typ	Hrs/wk	CP
Cybersecurity Data Science (L2914)	Lecture	2	3
Exercise Cybersecurity Data Science (L2915)	Project-/problem-based Learning	2	3

Module Responsible	Prof. Riccardo Scandariato
Admission Requirements	None
Recommended Previous Knowledge	Basic knowledge of probabilities and statistics. Familiarity with object oriented programming.
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students can: • Apply data science methods to the resolution of complex cybersecurity problems. • Use of data science methods to quantify risks and optimize cybersecurity operations. • Identify strengths and limitations of state-of-the-art methods • Select the performance indicators of data-oriented cybersecurity solutions. • Understand cybersecurity threats in data science methods.
Skills	Implement and evaluate data-driven models for the identification, treatment, and mitigation of cybersecurity risks
Personal Competence	
Social Competence	None
Autonomy	Students can apply the knowledge acquired throughout the course to the resolution of industrial case studies. Students should also be capable to acquire new knowledge independently from academic publications, techical standards, and white papers.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	120 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems: Elective Compulsory

Course L2914: Cybersecurity Data Science

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Riccardo Scandariato
Language	EN
Cycle	SoSe
Content	Theoretical Foundations: • Introduction to data science • Supervised and unsupervised learning • Data science methods (e.g., clustering, decision trees, artificial neural networks) • Performance metrics Cybersecutrity Applications: • Spam detection • Phishing detection • Intrusion detection • Access-control prediction • Denial of Service (DoS) prediction • Vulnerability/malware prediction • Adversarial machine learning
Literature	[1] Sarker, I.H., Kayes, A.S.M., Badsha, S., Alqahtani, H., Watters, P. and Ng, A., 2020. Cybersecurity data science: an overview from machine learning perspective. Journal of Big data, 7(1), pp.1-29. [2] Truong, T.C., Zelinka, I., Plucar, J., Čandík, M. and Šulc, V., 2020. Artificial intelligence and cybersecurity: Past, presence, and future. In Artificial intelligence and evolutionary computations in engineering systems (pp. 351-363). Springer, Singapore. [3] Dua, S. and Du, X., 2016. Data mining and machine learning in cybersecurity. CRC press. [4] Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L. and Rieck, K., Dos and Don'ts of Machine Learning in Computer Security. [5] Torres, J.M., Comesaña, C.I. and Garcia-Nieto, P.J., 2019. Machine learning techniques applied to cybersecurity. International Journal of Machine Learning and Cybernetics, 10(10), pp.2823-2836. [6] Russell, S. and Norvig, P., 2010. Artificial Intelligence: A Modern Approach, Prentice Hall.

<table>
<tr><th colspan="2">Course L2915: Exercise Cybersecurity Data Science</th></tr>
<tr><td>Typ</td><td>Project-/problem-based Learning</td></tr>
<tr><td>Hrs/wk</td><td>2</td></tr>
<tr><td>CP</td><td>3</td></tr>
<tr><td>Workload in Hours</td><td>Independent Study Time 62, Study Time in Lecture 28</td></tr>
<tr><td>Lecturer</td><td>Prof. Riccardo Scandariato</td></tr>
<tr><td>Language</td><td>EN</td></tr>
<tr><td>Cycle</td><td>SoSe</td></tr>
<tr><td>Content</td><td>Theoretical Foundations:
• Introduction to data science
• Supervised and unsupervised learning
• Data science methods (e.g., clustering, decision trees, artificial neural networks)
• Performance metrics

Cybersecutrity Applications:
• Spam detection
• Phishing detection
• Intrusion detection
• Access-control prediction
• Denial of Service (DoS) prediction
• Vulnerability/malware prediction
• Adversarial machine learning</td></tr>
<tr><td>Literature</td><td>[1] Sarker, I.H., Kayes, A.S.M., Badsha, S., Alqahtani, H., Watters, P. and Ng, A., 2020. Cybersecurity data science: an overview from machine learning perspective. Journal of Big data, 7(1), pp.1-29.

[2] Truong, T.C., Zelinka, I., Plucar, J., Čandík, M. and Šulc, V., 2020. Artificial intelligence and cybersecurity: Past, presence, and future. In Artificial intelligence and evolutionary computations in engineering systems (pp. 351-363). Springer, Singapore.

[3] Dua, S. and Du, X., 2016. Data mining and machine learning in cybersecurity. CRC press.

[4] Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L. and Rieck, K., Dos and Don'ts of Machine Learning in Computer Security.

[5] Torres, J.M., Comesaña, C.I. and Garcia-Nieto, P.J., 2019. Machine learning techniques applied to cybersecurity. International Journal of Machine Learning and Cybernetics, 10(10), pp.2823-2836.

[6] Russell, S. and Norvig, P., 2010. Artificial Intelligence: A Modern Approach, Prentice Hall.</td></tr>
</table>

Module M0924: Software for Embedded Systems

Courses

Title	Typ	Hrs/wk	CP
Software for Embdedded Systems (L1069)	Lecture	2	3
Software for Embdedded Systems (L1070)	Recitation Section (small)	3	3

Module Responsible	Prof. Bernd-Christian Renner
Admission Requirements	None
Recommended Previous Knowledge	• Very Good knowledge and practical experience in programming in the C language • Basic knowledge in software engineering • Basic understanding of assembly language
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students know the basic principles and procedures of software engineering for embedded systems. They are able to describe the usage and pros of event based programming using interrupts. They know the components and functions of a concrete microcontroller. The participants explain requirements of real time systems. They know at least three scheduling algorithms for real time operating systems including their pros and cons.
Skills	Students build interrupt-based programs for a concrete microcontroller. They build and use a preemptive scheduler. They use peripheral components (timer, ADC, EEPROM) to realize complex tasks for embedded systems. To interface with external components they utilize serial protocols.
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	**Compulsory** No, **Bonus** 10 %, **Form** Attestation, **Description**
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Electrical Engineering: Specialisation Information and Communication Systems: Elective Compulsory Information and Communication Systems: Specialisation Communication Systems, Focus Software: Elective Compulsory Mechatronics: Technical Complementary Course: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Mechatronics: Specialisation System Design: Elective Compulsory Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory

Course L1069: Software for Embdedded Systems

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Bernd-Christian Renner
Language	DE/EN
Cycle	SoSe
Content	• General-Purpose Processors • Programming the Atmel AVR • Interrupts • C for Embedded Systems • Standard Single Purpose Processors: Peripherals • Finite-State Machines • Memory • Operating Systems for Embedded Systems • Real-Time Embedded Systems • Boot loader and Power Management
Literature	1. Embedded System Design, F. Vahid and T. Givargis, John Wiley 2. Programming Embedded Systems: With C and Gnu Development Tools, M. Barr and A. Massa, O'Reilly 3. C und C++ für Embedded Systems, F. Bollow, M. Homann, K. Köhn, MITP 4. The Art of Designing Embedded Systems, J. Ganssle, Newnses 5. Mikrocomputertechnik mit Controllern der Atmel AVR-RISC-Familie, G. Schmitt, Oldenbourg 6. Making Embedded Systems: Design Patterns for Great Software, E. White, O'Reilly

Course L1070: Software for Embdedded Systems	
Typ	Recitation Section (small)
Hrs/wk	3
CP	3
Workload in Hours	Independent Study Time 48, Study Time in Lecture 42
Lecturer	Prof. Bernd-Christian Renner
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M1810: Autonomous Cyber-Physical Systems

Courses

Title	Typ	Hrs/wk	CP
Autonomous Cyber-Physical Systems (L3000)	Lecture	2	3
Autonomous Cyber-Physical Systems (L3001)	Recitation Section (small)	2	3

Module Responsible	Prof. Bernd-Christian Renner
Admission Requirements	None
Recommended Previous Knowledge	• Very Good knowledge and practical experience in programming in the C language (Module: Procedural Programming) • Basic knowledge in software engineering • Basic knowledge in wired and wireless communication protocols • Principal understanding of simple electronic circuits
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory** No · **Bonus** 10 % · **Form** Attestation · **Description**
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems, Focus Software and Signal Processing: Elective Compulsory

Course L3000: Autonomous Cyber-Physical Systems

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Bernd-Christian Renner
Language	EN
Cycle	SoSe
Content	
Literature	

Course L3001: Autonomous Cyber-Physical Systems

Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Bernd-Christian Renner
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M1812: Constraint Satisfaction Problems

Courses

Title	Typ	Hrs/wk	CP
Constraint Satisfaction Problems (L3002)	Lecture	2	3
Constraint Satisfaction Problems (L3003)	Recitation Section (large)	2	3

Module Responsible	Prof. Antoine Wiehe
Admission Requirements	None
Recommended Previous Knowledge	The students should have followed the courses Complexity Theory, Discrete Algebraic Structures, Linear Algebra.
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students can describe basic concepts from the theory of constraint satisfaction such as primitive positive formulas, interpretations, polymorphisms, clones • Students can discuss the connections between these concepts • Students know proofs strategies and can reproduce them
Skills	• Students can use CSPs to model problems from complexity theory and decide their complexity using methods from the course.
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory

Course L3002: Constraint Satisfaction Problems

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Antoine Wiehe
Language	EN
Cycle	SoSe
Content	This course gives an introduction to the topic of constraint satisfaction problems and their complexity. It will cover the basics of the theory such as the universal-algebraic approach to constraint satisfaction and several classical algorithms such as local consistency checking and the Bulatov-Dalmau algorithm. We will finally discuss the recent research directions in the field.
Literature	

Course L3003: Constraint Satisfaction Problems

Typ	Recitation Section (large)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Antoine Wiehe
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M1772: Smart Sensors

Courses

Title	Typ	Hrs/wk	CP
Smart Sensors (L2904)	Lecture	2	2
Smart Sensors Lab (L2905)	Project-/problem-based Learning	3	4

Module Responsible	Prof. Ulf Kulau
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	25 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory

Course L2904: Smart Sensors

Typ	Lecture
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Ulf Kulau
Language	DE/EN
Cycle	SoSe
Content	
Literature	

Course L2905: Smart Sensors Lab

Typ	Project-/problem-based Learning
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Ulf Kulau
Language	DE/EN
Cycle	SoSe
Content	
Literature	

Module M1794: Applied Cryptography

Courses			
Title	**Typ**	**Hrs/wk**	**CP**
Applied Cryptography (L2954)	Lecture	3	4
Applied Cryptography (L2955)	Recitation Section (small)	1	2

Module Responsible	Prof. Sibylle Fröschle
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory** No; **Bonus** 10 %; **Form** Excercises; **Description** Die Übungsaufgaben finden semesterbegleitend statt
Examination	Written exam
Examination duration and scale	120 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Information and Communication Systems: Specialisation Communication Systems, Focus Software: Elective Compulsory

Course L2954: Applied Cryptography

Typ	Lecture
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Sibylle Fröschle
Language	EN
Cycle	SoSe
Content	This module provides a comprehensive knowledge in modern cryptography and how it plays a key role in securing the digital world we live in today. We will thoroughly treat cryptographic primitives such as symmetric and asymmetric encryption schemes, cryptographic hash functions, message authentication codes, and digital signatures. Moreover, we will cover aspects of practical deployment such as key management, public key infrastructures, and secure storage of keys. We will see how everything comes together in applications such as the ubiquitous security protocols of the Internet (e.g. TLS and WPA3) and/or the Internet-of-things. We also discuss current challenges such as the need for post-quantum cryptography.
Literature	Introduction to Modern Cryptography, Third Edition, Jonathan Katz and Jehuda Lindell, Chapman & Hall/CRC, 2021 Sicherheit und Kryptographie im Internet, 5th Edition, Jörg Schwenk, Springer-Verlag, 2020

Course L2955: Applied Cryptography

Typ	Recitation Section (small)
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Sibylle Fröschle
Language	EN
Cycle	SoSe
Content	See corresponding lecture
Literature	Siehe korrespondierende Vorlesung

Module M1842: GPU Architectures

Courses			
Title	**Typ**	**Hrs/wk**	**CP**
GPU Architecture (L3039)	Lecture	3	4
GPU Architecture (L3040)	Project-/problem-based Learning	1	2

Module Responsible	Prof. Sohan Lal
Admission Requirements	None
Recommended Previous Knowledge	An introductory module on computer engineering or computer architecture, and good programming skills in C/C++.
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems, Focus Software and Signal Processing: Elective Compulsory Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory

Course L3039: GPU Architecture

Typ	Lecture
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Sohan Lal
Language	EN
Cycle	SoSe
Content	- Review of computer architecture basics - measuring performance, benchmarks, five-stage RISC pipeline, caches - GPU basics - evolution of GPU computing, a high-level overview of a GPU architecture - GPU programming with CUDA - program structure, CUDA threads organization, warp/thread-block scheduling - GPU (micro) architecture - streaming multiprocessors, single instruction multiple threads (SIMT) core design, tensor/RT cores, mixed-precision support - GPU memory hierarchy - banked register file and operand collectors, shared memory, GPU caches (differences w.r.t. CPU caches), global memory - Branch and memory divergence - branch handling, stack-based reconvergence, memory coalescing, coalescer design - Barriers and synchronization - Temporal and spatial locality exploitation challenges in GPU caches - Global memory- high throughput requirements, GDDR/HBM, memory bandwidth optimization techniques - GPU research issues - performance bottlenecks, GPU power modeling, high-power consumption/energy efficiency, GPU security - Application case study - deep learning - Cycle accurate simulators for GPUs The learning in the lectures will be augmented by a semester-long problem-based project.
Literature	

Course L3040: GPU Architecture	
Typ	Project-/problem-based Learning
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Sohan Lal
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M0839: Traffic Engineering

Courses			
Title	**Typ**	**Hrs/wk**	**CP**
Seminar Traffic Engineering (L0902)	Seminar	2	2
Traffic Engineering (L0900)	Lecture	2	2
Traffic Engineering Exercises (L0901)	Recitation Section (small)	1	2

Module Responsible	Prof. Andreas Timm-Giel
Admission Requirements	None
Recommended Previous Knowledge	• Fundamentals of communication or computer networks • Stochastics
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students are able to describe methods for planning, optimisation and performance evaluation of communication networks.
Skills	Students are able to solve typical planning and optimisation tasks for communication networks. Furthermore they are able to evaluate the network performance using queuing theory. Students are able to apply independently what they have learned to other and new problems. They can present their results in front of experts and discuss them.
Personal Competence	
Social Competence	
Autonomy	Students are able to acquire the necessary expert knowledge to understand the functionality and performance of new communication networks independently.
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Electrical Engineering: Specialisation Information and Communication Systems: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems, Focus Networks: Elective Compulsory

Course L0902: Seminar Traffic Engineering

Typ	Seminar
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Andreas Timm-Giel, Dr. Phuong Nga Tran
Language	EN
Cycle	WiSe
Content	Selected applications of methods for planning, optimization, and performance evaluation of communication networks, which have been introduced in the traffic engineering lecture are prepared by the students and presented in a seminar.
Literature	• U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Vieweg + Teubner • further literature announced in the lecture

Course L0900: Traffic Engineering	
Typ	Lecture
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Andreas Timm-Giel, Dr. Phuong Nga Tran
Language	EN
Cycle	WiSe
Content	Network Planning and Optimization • Linear Programming (LP) • Network planning with LP solvers • Planning of communication networks Queueing Theory for Communication Networks • Stochastic processes • Queueing systems • Switches (circuit- and packet switching) • Network of queues
Literature	Literatur: U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Springer Weitere Literatur wird in der Lehrveranstaltung bekanntgegeben / Literature: U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Springer further literature announced in the lecture

Course L0901: Traffic Engineering Exercises	
Typ	Recitation Section (small)
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Andreas Timm-Giel
Language	EN
Cycle	WiSe
Content	Accompanying exercise for the traffic engineering course
Literature	Literatur: U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Springer Weitere Literatur wird in der Lehrveranstaltung bekanntgegeben / Literature: U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Springer further literature announced in the lecture

Module M1742: Operating System Techniques

Courses

Title	Typ	Hrs/wk	CP
Operating System Techniques (L2815)	Lecture	1	2
Operating System Techniques (L2816)	Project-/problem-based Learning	3	4

Module Responsible	Prof. Christian Dietrich
Admission Requirements	None
Recommended Previous Knowledge	• Object-oriented programming (mandatory) • Programming in C/C++ (mandatory) • Operating system construction (recommended) • Basics of computer architecture (recommended)
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students who have successfully completed the module: • explain and implement design principles for system calls and discuss their specific advantages/disadvantages. • classify protection, management, and virtualization techniques for memory (paging, segementation, language-based, capabilities) and implement them on the IA-32 architecture. • compare basic OS architectures (monolith, microkernel, macrokernel, exokernel) on the basis of fundamental characteristics (robustness, performance, portability) and their influence on the implementation of mechanisms (system calls, address space protection). • discuss address space models (multi-address space model, single-address space model, multi-level and inverse page mappings, sharing) and their implementability on common hardware architectures. • discuss principles of code and data sharing with respect to operating system and address space architecture. • can distinguish logical, virtual, and physical memory. • can derive the cost advantages of zero-copy interfaces • can distinguish technical and conceptual views of process generation by fork().
Skills	Students who have successfully completed the module: • explain and implement design principles for system calls and discuss their specific advantages/disadvantages. • can implement basic mechanisms for memory management • understand typical problems (concurrency, compiler behavior, debugging without dedicated tools) and sources of errors in low-level software development. • are able to design basic abstractions for address space virtualization. • can name the necessary prerequisites for privilege separation and also implement these technically • implement techniques for lazy decoupling of memory operations (Copy-On Write) • implement mechanisms and abstractions for interprocess communication.
Personal Competence	
Social Competence	Students who have successfully completed the module: • can work cooperatively in small groups. • can present and argue their design and implementation decisions in a compact manner.
Autonomy	Students who have successfully completed the module: • are able to gradually understand complex error patterns by means of a methodical approach. • reflect critically on their design decisions and derive suitable alternatives. • can deal openly and constructively with weak points and wrong decisions. • can revise wrong decisions and/or accept the resulting costs only consciously. • can implement an abstract tasks in a goal-oriented manner.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	25 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory

Course L2815: Operating System Techniques	
Typ	Lecture
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Christian Dietrich
Language	DE/EN
Cycle	WiSe
Content	The main focus of the course is the management of virtual address spaces. We examine methods and techniques for separating logical address spaces, for accessing memory across address-space boundaries and for isolating processes. We also explore the implementation of system calls and as well as page- and segment-based techniques for mapping logical/virtual address spaces to physical memory. With this background, different operating system architectures are compared and common address space models of operating systems are explained. Further topics are interprocess communication by message passing in case of separated address spaces, but also the replication of virtual shared memory based on these techniques. The lecture provides the necessary knowledge to extend a given micro operating system with memory protection and privilege isolation.
Literature	

Course L2816: Operating System Techniques	
Typ	Project-/problem-based Learning
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Christian Dietrich
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1780: Massively Parallel Systems: Architecture and Programming

Courses

Title	Typ	Hrs/wk	CP
Massively Parallel Systems: Architecture and Programming (L2936)	Lecture	2	3
Massively Parallel Systems: Architecture and Programming (L2937)	Project-/problem-based Learning	2	3

Module Responsible	Prof. Sohan Lal
Admission Requirements	None
Recommended Previous Knowledge	An introductory module on computer Engineering or computer architecture, good programming skills in C/C++.
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	The course starts with parallel computers classification, multithreading, and covers the architecture of centralized and distributed shared-memory parallel systems, multiprocessor cache coherence, snooping / directory-based cache coherence protocols, implementation, and limitations. Next, students study interconnection networks and routing in parallel systems. To ensure the correctness of shared-memory multithreaded programs, independent of the speed of execution of their individual threads, the important topics of memory consistency and synchronization will be covered in detail. As a case study, the architecture of a few accelerators such as GPUs will also be discussed in detail. Besides understanding the architecture and organization of parallel systems, programming them is also very challenging. The course will also cover how to program massively parallel systems using API/libraries such as CUDA/OpenCL/MPI/OpenMP.
Skills	After completing this course, students will be able to understand the architecture and organization of parallel systems. They will be able to evaluate different design choices and make decisions while designing a parallel system. In addition, they will be able to program parallel systems (ranging from an embedded system to a supercomputer) using CUDA/OpenCL/MPI/OpenMP.
Personal Competence	
Social Competence	The course will encourage students to work in small groups to solve complex problems, thus, inculcating the importance of teamwork.
Autonomy	Today, parallel computers are present everywhere. Students will be able to not only program parallel computers independently, but also understand their underlying organization and architecture. This will further help to understand the performance issues of parallel applications and provide insights to improve them.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory** Yes; **Bonus** 20 %; **Form** Subject theoretical and practical work; **Description**
Examination	Oral exam
Examination duration and scale	25 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Data Science: Specialisation II. Computer Science: Elective Compulsory Data Science: Specialisation IV. Special Focus Area: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Information and Communication Systems: Specialisation Communication Systems, Focus Software: Elective Compulsory Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory

Course L2936: Massively Parallel Systems: Architecture and Programming

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sohan Lal
Language	EN
Cycle	WiSe
Content	Brief outline: • Parallel computers and their classification • Centralized and distributed shared-memory architectures: snooping vs directory-based cache coherence protocols, implementation, and limitations • Chip multiprocessors: software-based, block (coarse-grain), interleaved (fine-grain), simultaneous multithreading • Synchronization: high-level primitives and implementation, memory consistency models: sequential and weaker memory consistency models • Interconnection networks: topologies (direct and indirect networks) and routing techniques • Graphics Processing Units (GPUs) architecture and programming using CUDA/OpenCL • Parallel programming with message passing interface (MPI), OpenMP
Literature	• Michel Dubois, Murali Annavaram, and Per Stenström, Parallel Computer Organization and Design (Book) • David A Patterson and John L. Hennessy, Computer Architecture: A Quantitative Approach, Elsevier (Book) • David B. Kirk, Wen-mei W. Hwu, Programming Massivley Parallel Processors, Elsevier (Book)

Course L2937: Massively Parallel Systems: Architecture and Programming	
Typ	Project-/problem-based Learning
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sohan Lal
Language	EN
Cycle	WiSe
Content	There will be 3-4 assignments for project-based learning consisting of the following: • Implement and compare different cache coherence protocols using a simulator or a high-level, event-driven simulation interface such as SystemC • Programming massively parallel systems to solve computationally intensive problems such as password cracking using CUDA/OpenCL/MPI/OpenMP
Literature	The following literature will be useful for project-based learning. The further required resources will be discussed during the course. • David B. Kirk, Wen-mei W. Hwu, Programming Massivley Parallel Processors, Elsevier (Book) • MPI Forum, https://www.mpi-forum.org/ • SystemC, https://www.accellera.org/community/systemc

Specialization II: Intelligence Engineering

Module M0633: Industrial Process Automation

Courses

Title	Typ	Hrs/wk	CP
Industrial Process Automation (L0344)	Lecture	2	3
Industrial Process Automation (L0345)	Recitation Section (small)	2	3

Module Responsible	Prof. Alexander Schlaefer
Admission Requirements	None
Recommended Previous Knowledge	mathematics and optimization methods principles of automata principles of algorithms and data structures programming skills
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	The students can evaluate and assess discrete event systems. They can evaluate properties of processes and explain methods for process analysis. The students can compare methods for process modelling and select an appropriate method for actual problems. They can discuss scheduling methods in the context of actual problems and give a detailed explanation of advantages and disadvantages of different programming methods. The students can relate process automation to methods from robotics and sensor systems as well as to recent topics like 'cyberphysical systems' and 'industry 4.0'.
Skills	The students are able to develop and model processes and evaluate them accordingly. This involves taking into account optimal scheduling, understanding algorithmic complexity, and implementation using PLCs.
Personal Competence	
Social Competence	The students can independently define work processes within their groups, distribute tasks within the group and develop solutions collaboratively.
Autonomy	The students are able to assess their level of knowledge and to document their work results adequately.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory** No · **Bonus** 10 % · **Form** Excercises · **Description**
Examination	Written exam
Examination duration and scale	90 minutes
Assignment for the Following Curricula	Bioprocess Engineering: Specialisation A - General Bioprocess Engineering: Elective Compulsory Chemical and Bioprocess Engineering: Specialisation Chemical Process Engineering: Elective Compulsory Chemical and Bioprocess Engineering: Specialisation General Process Engineering: Elective Compulsory Computer Science: Specialisation II: Intelligence Engineering: Elective Compulsory Electrical Engineering: Specialisation Control and Power Systems Engineering: Elective Compulsory Aircraft Systems Engineering: Core Qualification: Elective Compulsory International Management and Engineering: Specialisation II. Mechatronics: Elective Compulsory International Management and Engineering: Specialisation II. Product Development and Production: Elective Compulsory Mechanical Engineering and Management: Specialisation Mechatronics: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory Process Engineering: Specialisation Chemical Process Engineering: Elective Compulsory Process Engineering: Specialisation Process Engineering: Elective Compulsory

Course L0344: Industrial Process Automation	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Alexander Schlaefer
Language	EN
Cycle	WiSe
Content	- foundations of problem solving and system modeling, discrete event systems - properties of processes, modeling using automata and Petri-nets - design considerations for processes (mutex, deadlock avoidance, liveness) - optimal scheduling for processes - optimal decisions when planning manufacturing systems, decisions under uncertainty - software design and software architectures for automation, PLCs
Literature	J. Lunze: „Automatisierungstechnik“, Oldenbourg Verlag, 2012 Reisig: Petrinetze: Modellierungstechnik, Analysemethoden, Fallstudien; Vieweg+Teubner 2010 Hrúz, Zhou: Modeling and Control of Discrete-event Dynamic Systems; Springer 2007 Li, Zhou: Deadlock Resolution in Automated Manufacturing Systems, Springer 2009 Pinedo: Planning and Scheduling in Manufacturing and Services, Springer 2009

Course L0345: Industrial Process Automation	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Alexander Schlaefer
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M0629: Intelligent Autonomous Agents and Cognitive Robotics

Courses

Title	Typ	Hrs/wk	CP
Intelligent Autonomous Agents and Cognitive Robotics (L0341)	Lecture	2	4
Intelligent Autonomous Agents and Cognitive Robotics (L0512)	Recitation Section (small)	2	2

Module Responsible	Rainer Marrone
Admission Requirements	None
Recommended Previous Knowledge	Vectors, matrices, Calculus
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students can explain the agent abstraction, define intelligence in terms of rational behavior, and give details about agent design (goals, utilities, environments). They can describe the main features of environments. The notion of adversarial agent cooperation can be discussed in terms of decision problems and algorithms for solving these problems. For dealing with uncertainty in real-world scenarios, students can summarize how Bayesian networks can be employed as a knowledge representation and reasoning formalism in static and dynamic settings. In addition, students can define decision making procedures in simple and sequential settings, with and with complete access to the state of the environment. In this context, students can describe techniques for solving (partially observable) Markov decision problems, and they can recall techniques for measuring the value of information. Students can identify techniques for simultaneous localization and mapping, and can explain planning techniques for achieving desired states. Students can explain coordination problems and decision making in a multi-agent setting in term of different types of equilibria, social choice functions, voting protocol, and mechanism design techniques.
Skills	Students can select an appropriate agent architecture for concrete agent application scenarios. For simplified agent application students can derive decision trees and apply basic optimization techniques. For those applications they can also create Bayesian networks/dynamic Bayesian networks and apply bayesian reasoning for simple queries. Students can also name and apply different sampling techniques for simplified agent scenarios. For simple and complex decision making students can compute the best action or policies for concrete settings. In multi-agent situations students will apply techniques for finding different equilibria states,e.g., Nash equilibria. For multi-agent decision making students will apply different voting protocols and compare and explain the results.
Personal Competence	
Social Competence	Students are able to discuss their solutions to problems with others. They communicate in English
Autonomy	Students are able of checking their understanding of complex concepts by solving varaints of concrete problems
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 minutes
Assignment for the Following Curricula	Computer Science: Specialisation II: Intelligence Engineering: Elective Compulsory International Management and Engineering: Specialisation II. Information Technology: Elective Compulsory Mechatronics: Technical Complementary Course: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Biomedical Engineering: Specialisation Artificial Organs and Regenerative Medicine: Elective Compulsory Biomedical Engineering: Specialisation Implants and Endoprostheses: Elective Compulsory Biomedical Engineering: Specialisation Medical Technology and Control Theory: Elective Compulsory Biomedical Engineering: Specialisation Management and Business Administration: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory

Course L0341: Intelligent Autonomous Agents and Cognitive Robotics	
Typ	Lecture
Hrs/wk	2
CP	4
Workload in Hours	Independent Study Time 92, Study Time in Lecture 28
Lecturer	Rainer Marrone
Language	EN
Cycle	WiSe
Content	• Definition of agents, rational behavior, goals, utilities, environment types • Adversarial agent cooperation: Agents with complete access to the state(s) of the environment, games, Minimax algorithm, alpha-beta pruning, elements of chance • Uncertainty: Motivation: agents with no direct access to the state(s) of the environment, probabilities, conditional probabilities, product rule, Bayes rule, full joint probability distribution, marginalization, summing out, answering queries, complexity, independence assumptions, naive Bayes, conditional independence assumptions • Bayesian networks: Syntax and semantics of Bayesian networks, answering queries revised (inference by enumeration), typical-case complexity, pragmatics: reasoning from effect (that can be perceived by an agent) to cause (that cannot be directly perceived). • Probabilistic reasoning over time: Environmental state may change even without the agent performing actions, dynamic Bayesian networks, Markov assumption, transition model, sensor model, inference problems: filtering, prediction, smoothing, most-likely explanation, special cases: hidden Markov models, Kalman filters, Exact inferences and approximations • Decision making under uncertainty: Simple decisions: utility theory, multivariate utility functions, dominance, decision networks, value of informatio Complex decisions: sequential decision problems, value iteration, policy iteration, MDPs Decision-theoretic agents: POMDPs, reduction to multidimensional continuous MDPs, dynamic decision networks • Simultaneous Localization and Mapping • Planning • Game theory (Golden Balls: Split or Share) Decisions with multiple agents, Nash equilibrium, Bayes-Nash equilibrium • Social Choice Voting protocols, preferences, paradoxes, Arrow's Theorem, • Mechanism Design Fundamentals, dominant strategy implementation, Revelation Principle, Gibbard-Satterthwaite Impossibility Theorem, Direct mechanisms, incentive compatibility, strategy-proofness, Vickrey-Groves-Clarke mechanisms, expected externality mechanisms, participation constraints, individual rationality, budget balancedness, bilateral trade, Myerson-Satterthwaite Theorem
Literature	1. Artificial Intelligence: A Modern Approach (Third Edition), Stuart Russell, Peter Norvig, Prentice Hall, 2010, Chapters 2-5, 10-11, 13-17 2. Probabilistic Robotics, Thrun, S., Burgard, W., Fox, D. MIT Press 2005 3. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Yoav Shoham, Kevin Leyton-Brown, Cambridge University Press, 2009

Course L0512: Intelligent Autonomous Agents and Cognitive Robotics	
Typ	Recitation Section (small)
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Rainer Marrone
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M0630: Robotics and Navigation in Medicine

Courses

Title	Typ	Hrs/wk	CP
Robotics and Navigation in Medicine (L0335)	Lecture	2	3
Robotics and Navigation in Medicine (L0338)	Project Seminar	2	2
Robotics and Navigation in Medicine (L0336)	Recitation Section (small)	1	1

Module Responsible	Prof. Alexander Schlaefer
Admission Requirements	None
Recommended Previous Knowledge	• principles of math (algebra, analysis/calculus) • principles of programming, e.g., in Java or C++ • solid R or Matlab skills
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	The students can explain kinematics and tracking systems in clinical contexts and illustrate systems and their components in detail. Systems can be evaluated with respect to collision detection and safety and regulations. Students can assess typical systems regarding design and limitations.
Skills	The students are able to design and evaluate navigation systems and robotic systems for medical applications.
Personal Competence	
Social Competence	The students are able to grasp practical tasks in groups, develop solution strategies independently, define work processes and work on them collaboratively. The students are able to collaboratively organize their work processes and software solutions using virtual communication and software management tools. The students can critically reflect on the results of other groups, make constructive suggestions for improvement, and also incorporate them into their own work.
Autonomy	The students can assess their level of knowledge and independently control their learning processes on this basis as well as document their work results. They can critically evaluate the results achieved and present them in an appropriate argumentative manner to the other groups.
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	**Compulsory** / **Bonus** / **Form** / **Description** Yes / 10 % / Presentation Yes / 10 % / Written elaboration
Examination	Written exam
Examination duration and scale	90 minutes
Assignment for the Following Curricula	Computer Science: Specialisation II: Intelligence Engineering: Elective Compulsory Electrical Engineering: Specialisation Medical Technology: Elective Compulsory International Management and Engineering: Specialisation II. Electrical Engineering: Elective Compulsory International Management and Engineering: Specialisation II. Process Engineering and Biotechnology: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Biomedical Engineering: Specialisation Artificial Organs and Regenerative Medicine: Elective Compulsory Biomedical Engineering: Specialisation Implants and Endoprostheses: Elective Compulsory Biomedical Engineering: Specialisation Medical Technology and Control Theory: Elective Compulsory Biomedical Engineering: Specialisation Management and Business Administration: Elective Compulsory Product Development, Materials and Production: Specialisation Product Development: Elective Compulsory Product Development, Materials and Production: Specialisation Production: Elective Compulsory Product Development, Materials and Production: Specialisation Materials: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Bio- and Medical Technology: Elective Compulsory

Course L0335: Robotics and Navigation in Medicine	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Alexander Schlaefer
Language	EN
Cycle	SoSe
Content	- kinematics - calibration - tracking systems - navigation and image guidance - motion compensation The seminar extends and complements the contents of the lecture with respect to recent research results.
Literature	Spong et al.: Robot Modeling and Control, 2005 Troccaz: Medical Robotics, 2012 Further literature will be given in the lecture.

Course L0338: Robotics and Navigation in Medicine	
Typ	Project Seminar
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Alexander Schlaefer
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Course L0336: Robotics and Navigation in Medicine	
Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Alexander Schlaefer
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M0627: Machine Learning and Data Mining

Courses

Title	Typ	Hrs/wk	CP
Machine Learning and Data Mining (L0340)	Lecture	2	4
Machine Learning and Data Mining (L0510)	Recitation Section (small)	2	2

Module Responsible	NN
Admission Requirements	None
Recommended Previous Knowledge	• Calculus • Stochastics
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students can explain the difference between instance-based and model-based learning approaches, and they can enumerate basic machine learning technique for each of the two basic approaches, either on the basis of static data, or on the basis of incrementally incoming data . For dealing with uncertainty, students can describe suitable representation formalisms, and they explain how axioms, features, parameters, or structures used in these formalisms can be learned automatically with different algorithms. Students are also able to sketch different clustering techniques. They depict how the performance of learned classifiers can be improved by ensemble learning, and they can summarize how this influences computational learning theory. Algorithms for reinforcement learning can also be explained by students.
Skills	Student derive decision trees and, in turn, propositional rule sets from simple and static data tables and are able to name and explain basic optimization techniques. They present and apply the basic idea of first-order inductive leaning. Students apply the BME, MAP, ML, and EM algorithms for learning parameters of Bayesian networks and compare the different algorithms. They also know how to carry out Gaussian mixture learning. They can contrast kNN classifiers, neural networks, and support vector machines, and name their basic application areas and algorithmic properties. Students can describe basic clustering techniques and explain the basic components of those techniques. Students compare related machine learning techniques, e.g., k-means clustering and nearest neighbor classification. They can distinguish various ensemble learning techniques and compare the different goals of those techniques.
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 minutes
Assignment for the Following Curricula	Computer Science: Specialisation II: Intelligence Engineering: Elective Compulsory International Management and Engineering: Specialisation II. Information Technology: Elective Compulsory Mechatronics: Technical Complementary Course: Elective Compulsory Mechatronics: Specialisation System Design: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory

Course L0340: Machine Learning and Data Mining	
Typ	Lecture
Hrs/wk	2
CP	4
Workload in Hours	Independent Study Time 92, Study Time in Lecture 28
Lecturer	Rainer Marrone
Language	EN
Cycle	SoSe
Content	• Decision trees • First-order inductive learning • Incremental learning: Version spaces • Uncertainty • Bayesian networks • Learning parameters of Bayesian networks BME, MAP, ML, EM algorithm • Learning structures of Bayesian networks • Gaussian Mixture Models • kNN classifier, neural network classifier, support vector machine (SVM) classifier • Clustering Distance measures, k-means clustering, nearest neighbor clustering • Kernel Density Estimation • Ensemble Learning • Reinforcement Learning • Computational Learning Theory
Literature	1. Artificial Intelligence: A Modern Approach (Third Edition), Stuart Russel, Peter Norvig, Prentice Hall, 2010, Chapters 13, 14, 18-21 2. Machine Learning: A Probabilistic Perspective, Kevin Murphy, MIT Press 2012

Course L0510: Machine Learning and Data Mining	
Typ	Recitation Section (small)
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Rainer Marrone
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M1702: Process Imaging

Courses

Title	Typ	Hrs/wk	CP
Process Imaging (L2723)	Lecture	3	3
Process Imaging (L2724)	Project-/problem-based Learning	3	3

Module Responsible	Prof. Alexander Penn
Admission Requirements	None
Recommended Previous Knowledge	No special prerequisites needed
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	**Content:** The module focuses primarily on discussing established imaging techniques including (a) optical and infrared imaging, (b) magnetic resonance imaging, (c) X-ray imaging and tomography, and (d) ultrasound imaging but also covers a range of more recent imaging modalities. The students will learn: 1. what these imaging techniques can measure (such as sample density or concentration, material transport, chemical composition, temperature), 2. how the measurements work (physical measurement principles, hardware requirements, image reconstruction), and 3. how to determine the most suited imaging methods for a given problem. **Learning goals:** After the successful completion of the course, the students shall: 1. understand the physical principles and practical aspects of the most common imaging methods, 2. be able to assess the pros and cons of these methods with regard to cost, complexity, expected contrasts, spatial and temporal resolution, and based on this assessment 3. be able to identify the most suited imaging modality for any specific engineering challenge in the field of chemical and bioprocess engineering.
Skills	
Personal Competence	
Social Competence	In the problem-based interactive course, students work in small teams and set up two process imaging systems and use these systems to measure relevant process parameters in different chemical and bioprocess engineering applications. The teamwork will foster interpersonal communication skills.
Autonomy	Students are guided to work in self-motivation due to the challenge-based character of this module. A final presentation improves presentation skills.
Workload in Hours	Independent Study Time 96, Study Time in Lecture 84
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	120 min
Assignment for the Following Curricula	Bioprocess Engineering: Specialisation A - General Bioprocess Engineering: Elective Compulsory Bioprocess Engineering: Specialisation B - Industrial Bioprocess Engineering: Elective Compulsory Bioprocess Engineering: Specialisation C - Bioeconomic Process Engineering, Focus Energy and Bioprocess Technology: Elective Compulsory Chemical and Bioprocess Engineering: Specialisation General Process Engineering: Elective Compulsory Chemical and Bioprocess Engineering: Specialisation Bioprocess Engineering: Elective Compulsory Chemical and Bioprocess Engineering: Specialisation Chemical Process Engineering: Elective Compulsory Computer Science: Specialisation II: Intelligence Engineering: Elective Compulsory Information and Communication Systems: Specialisation Communication Systems, Focus Signal Processing: Elective Compulsory International Management and Engineering: Specialisation II. Process Engineering and Biotechnology: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory Process Engineering: Specialisation Process Engineering: Elective Compulsory Process Engineering: Specialisation Chemical Process Engineering: Elective Compulsory Process Engineering: Specialisation Environmental Process Engineering: Elective Compulsory Water and Environmental Engineering: Specialisation Environment: Elective Compulsory Water and Environmental Engineering: Specialisation Water: Elective Compulsory

Course L2723: Process Imaging	
Typ	Lecture
Hrs/wk	3
CP	3
Workload in Hours	Independent Study Time 48, Study Time in Lecture 42
Lecturer	Prof. Alexander Penn
Language	EN
Cycle	SoSe
Content	
Literature	Wang, M. (2015). Industrial Tomography. Cambridge, UK: Woodhead Publishing. Available as e-book in the library of TUHH: https://katalog.tub.tuhh.de/Record/823579395

Course L2724: Process Imaging	
Typ	Project-/problem-based Learning
Hrs/wk	3
CP	3
Workload in Hours	Independent Study Time 48, Study Time in Lecture 42
Lecturer	Prof. Alexander Penn, Dr. Stefan Benders
Language	EN
Cycle	SoSe
Content	**Content:** The module focuses primarily on discussing established imaging techniques including (a) optical and infrared imaging, (b) magnetic resonance imaging, (c) X-ray imaging and tomography, and (d) ultrasound imaging and also covers a range of more recent imaging modalities. The students will learn: 1. what these imaging techniques can measure (such as sample density or concentration, material transport, chemical composition, temperature), 2. how the measurements work (physical measurement principles, hardware requirements, image reconstruction), and 3. how to determine the most suited imaging methods for a given problem. **Learning goals:** After the successful completion of the course, the students shall: 1. understand the physical principles and practical aspects of the most common imaging methods, 2. be able to assess the pros and cons of these methods with regard to cost, complexity, expected contrasts, spatial and temporal resolution, and based on this assessment 3. be able to identify the most suited imaging modality for any specific engineering challenge in the field of chemical and bioprocess engineering.
Literature	Wang, M. (2015). Industrial Tomography. Cambridge, UK: Woodhead Publishing. Available as e-book in the library of TUHH: https://katalog.tub.tuhh.de/Record/823579395

Module M0623: Intelligent Systems in Medicine

Courses

Title	Typ	Hrs/wk	CP
Intelligent Systems in Medicine (L0331)	Lecture	2	3
Intelligent Systems in Medicine (L0334)	Project Seminar	2	2
Intelligent Systems in Medicine (L0333)	Recitation Section (small)	1	1

Module Responsible	Prof. Alexander Schlaefer
Admission Requirements	None
Recommended Previous Knowledge	• principles of math (algebra, analysis/calculus) • principles of stochastics • principles of programming, Java/C++ and R/Matlab • advanced programming skills
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	The students are able to analyze and solve clinical treatment planning and decision support problems using methods for search, optimization, and planning. They are able to explain methods for classification and their respective advantages and disadvantages in clinical contexts. The students can compare different methods for representing medical knowledge. They can evaluate methods in the context of clinical data and explain challenges due to the clinical nature of the data and its acquisition and due to privacy and safety requirements.
Skills	The students can give reasons for selecting and adapting methods for classification, regression, and prediction. They can assess the methods based on actual patient data and evaluate the implemented methods.
Personal Competence	
Social Competence	The students are able to grasp practical tasks in groups, develop solution strategies independently, define work processes and work on them collaboratively. The students can critically reflect on the results of other groups, make constructive suggestions for improvement and also incorporate them into their own work.
Autonomy	The students can assess their level of knowledge and document their work results. They can critically evaluate the results achieved and present them in an appropriate argumentative manner to the other groups.
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	**Compulsory** / **Bonus** / **Form** / **Description** Yes / 10 % / Written elaboration Yes / 10 % / Presentation
Examination	Written exam
Examination duration and scale	90 minutes
Assignment for the Following Curricula	Computer Science: Specialisation II: Intelligence Engineering: Elective Compulsory Data Science: Specialisation III. Applications: Elective Compulsory Data Science: Specialisation IV. Special Focus Area: Elective Compulsory Electrical Engineering: Specialisation Medical Technology: Elective Compulsory Interdisciplinary Mathematics: Specialisation Computational Methods in Biomedical Imaging: Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Mechatronics: Core Qualification: Elective Compulsory Biomedical Engineering: Specialisation Artificial Organs and Regenerative Medicine: Elective Compulsory Biomedical Engineering: Specialisation Implants and Endoprostheses: Elective Compulsory Biomedical Engineering: Specialisation Management and Business Administration: Elective Compulsory Biomedical Engineering: Specialisation Medical Technology and Control Theory: Compulsory Theoretical Mechanical Engineering: Specialisation Bio- and Medical Technology: Elective Compulsory

Course L0331: Intelligent Systems in Medicine	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Alexander Schlaefer
Language	EN
Cycle	WiSe
Content	- methods for search, optimization, planning, classification, regression and prediction in a clinical context - representation of medical knowledge - understanding challenges due to clinical and patient related data and data acquisition The students will work in groups to apply the methods introduced during the lecture using problem based learning.
Literature	Russel & Norvig: Artificial Intelligence: a Modern Approach, 2012 Berner: Clinical Decision Support Systems: Theory and Practice, 2007 Greenes: Clinical Decision Support: The Road Ahead, 2007 Further literature will be given in the lecture

Course L0334: Intelligent Systems in Medicine	
Typ	Project Seminar
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Alexander Schlaefer
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Course L0333: Intelligent Systems in Medicine	
Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Alexander Schlaefer
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1302: Applied Humanoid Robotics

Courses

Title	Typ	Hrs/wk	CP
Applied Humanoid Robotics (L1794)	Project-/problem-based Learning	6	6

Module Responsible	Patrick Göttsch
Admission Requirements	None
Recommended Previous Knowledge	• Object oriented programming; algorithms and data structures • Introduction to control systems • Control systems theory and design • Mechanics
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students can explain humanoid robots. • Students can explain the basic concepts, relationships and methods of forward- and inverse kinematics • Students learn to apply basic control concepts for different tasks in humanoid robotics.
Skills	• Students can implement models for humanoid robotic systems in Matlab and C++, and use these models for robot motion or other tasks. • They are capable of using models in Matlab for simulation and testing these models if necessary with C++ code on the real robot system. • They are capable of selecting methods for solving abstract problems, for which no standard methods are available, and apply it successfully.
Personal Competence	
Social Competence	• Students can develop joint solutions in mixed teams and present these. • They can provide appropriate feedback to others, and constructively handle feedback on their own results
Autonomy	• Students are able to obtain required information from provided literature sources, and to put in into the context of the lecture. • They can independently define tasks and apply the appropriate means to solve them.
Workload in Hours	Independent Study Time 96, Study Time in Lecture 84
Credit points	6
Course achievement	None
Examination	Written elaboration
Examination duration and scale	5-10 pages
Assignment for the Following Curricula	Computer Science: Specialisation II: Intelligence Engineering: Elective Compulsory Data Science: Specialisation III. Applications: Elective Compulsory Data Science: Specialisation IV. Special Focus Area: Elective Compulsory Electrical Engineering: Specialisation Control and Power Systems Engineering: Elective Compulsory Mechatronics: Core Qualification: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Bio- and Medical Technology: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory

Course L1794: Applied Humanoid Robotics

Typ	Project-/problem-based Learning
Hrs/wk	6
CP	6
Workload in Hours	Independent Study Time 96, Study Time in Lecture 84
Lecturer	Patrick Göttsch
Language	DE/EN
Cycle	WiSe/SoSe
Content	• Fundamentals of kinematics • Static and dynamic stability of humanoid robotic systems • Combination of different software environments (Matlab, C++, etc.) • Introduction to the necessary software frameworks • Team project • Presentation and Demonstration of intermediate and final results
Literature	• B. Siciliano, O. Khatib. "Handbook of Robotics. Part A: Robotics Foundations", Springer (2008)

Module M1249: Medical Imaging

Courses

Title	Typ	Hrs/wk	CP
Medical Imaging (L1694)	Lecture	2	3
Medical Imaging (L1695)	Recitation Section (small)	2	3

Module Responsible	Prof. Tobias Knopp
Admission Requirements	None
Recommended Previous Knowledge	Basic knowledge in linear algebra, numerics, and signal processing
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	After successful completion of the module, students are able to describe reconstruction methods for different tomographic imaging modalities such as computed tomography and magnetic resonance imaging. They know the necessary basics from the fields of signal processing and inverse problems and are familiar with both analytical and iterative image reconstruction methods. The students have a deepened knowledge of the imaging operators of computed tomography and magnetic resonance imaging.
Skills	The students are able to implement reconstruction methods and test them using tomographic measurement data. They can visualize the reconstructed images and evaluate the quality of their data and results. In addition, students can estimate the temporal complexity of imaging algorithms.
Personal Competence	
Social Competence	Students can work on complex problems both independently and in teams. They can exchange ideas with each other and use their individual strengths to solve the problem.
Autonomy	Students are able to independently investigate a complex problem and assess which competencies are required to solve it.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Computer Science: Specialisation II: Intelligence Engineering: Elective Compulsory Data Science: Specialisation III. Applications: Elective Compulsory Data Science: Specialisation IV. Special Focus Area: Elective Compulsory Electrical Engineering: Specialisation Medical Technology: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Interdisciplinary Mathematics: Specialisation Computational Methods in Biomedical Imaging: Compulsory Microelectronics and Microsystems: Specialisation Communication and Signal Processing: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Bio- and Medical Technology: Elective Compulsory

Course L1694: Medical Imaging

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Tobias Knopp
Language	DE/EN
Cycle	WiSe
Content	• Overview about different imaging methods • Signal processing • Inverse problems • Computed tomography • Magnetic resonance imaging • Compressed Sensing • Magnetic particle imaging
Literature	**Bildgebende Verfahren in der Medizin**; O. Dössel; Springer, Berlin, 2000 **Bildgebende Systeme für die medizinische Diagnostik**; H. Morneburg (Hrsg.); Publicis MCD, München, 1995 **Introduction to the Mathematics of Medical Imaging**; C. L.Epstein; Siam, Philadelphia, 2008 **Medical Image Processing, Reconstruction and Restoration**; J. Jan; Taylor and Francis, Boca Raton, 2006 **Principles of Magnetic Resonance Imaging**; Z.-P. Liang and P. C. Lauterbur; IEEE Press, New York, 1999

Course L1695: Medical Imaging	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Tobias Knopp
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Specialization III. Mathematics

Module M0667: Algorithmic Algebra

Courses

Title	Typ	Hrs/wk	CP
Algorithmic Algebra (L0422)	Lecture	3	5
Algorithmic Algebra (L0423)	Recitation Section (small)	1	1

Module Responsible	Dr. Prashant Batra
Admission Requirements	None
Recommended Previous Knowledge	Mathe I-III (Real analysis,computing in Vector spaces , principle of complete induction) Diskrete Mathematik I (gropus, rings, ideals, fields; euclidean algorithm)
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students can discuss logical connections between the following concepts and explain them by means of examples: Smith normal form, Chinese remainder theorem, grid point sets, integer solution of inequality systems.
Skills	Students are able to access independently further logical connections between the concepts with which they have become familiar and are able to verify them. Students are able to develop a suitable solution approach to given problems, to pursue it and to evaluate the results critically, such as in solving multivariate equation systems and in grid point theory.
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation III. Mathematics: Elective Compulsory

Course L0422: Algorithmic Algebra	
Typ	Lecture
Hrs/wk	3
CP	5
Workload in Hours	Independent Study Time 108, Study Time in Lecture 42
Lecturer	Dr. Prashant Batra
Language	DE
Cycle	WiSe
Content	Extended euclidean algorithm, solution of the Bezout-equation Division with remainder (over rings) fast arithmetic algorithms (conversion, fast multiplications) discrete Fourier-transformation over rings Computation with modular remainders, solving of remainder systems (chinese remainder theorem), solvability of integer linear systems over the integers linearization of polynomial equations-- matrix approach Sylvester-matrix, elimination elimination in rings, elimination of many variables Buchberger algorithm, Gröbner basis Minkowskis Lattice Point theorem and integer-valued optimization LLL-algorithm for construction of 'short' lattice vectors in polynomial time
Literature	von zur Gathen, Joachim; Gerhard, Jürgen Modern computer algebra. 3rd ed. (English) Zbl 1277.68002 Cambridge: Cambridge University Press (ISBN 978-1-107-03903-2/hbk; 978-1-139-85606-5/ebook). Yap, Chee Keng Fundamental problems of algorithmic algebra. (English) Zbl 0999.68261 Oxford: Oxford University Press. xvi, 511 p. $ 87.00 (2000). Free download for students from author's website: http://cs.nyu.edu/yap/book/berlin/ Cox, David; Little, John; O'Shea, Donal Ideals, varieties, and algorithms. An introduction to computational algebraic geometry and commutative algebra. 3rd ed. (English) Zbl 1118.13001 Undergraduate Texts in Mathematics. New York, NY: Springer (ISBN 978-0-387-35650-1/hbk; 978-0-387-35651-8/ebook). xv, 551 p. eBook: http://dx.doi.org/10.1007/978-0-387-35651-8 Concrete abstract algebra : from numbers to Gröbner bases / Niels **Lauritzen** **Verfasser:** **Lauritzen,** Niels **Ausgabe:** Reprinted with corr. **Erschienen:** Cambridge [u.a.] : Cambridge Univ. Press, 2006 **Umfang:** XIV, 240 S. : graph. Darst. **Anmerkung:** Includes bibliographical references and index **ISBN:** 0-521-82679-9, 978-0-521-82679-2 (hbk.) : GBP 55.00 0-521-53410-0, 978-0-521-53410-9 (pbk.) : USD 39.99 Koepf, Wolfram Computer algebra. An algorithmic oriented introduction. (Computeralgebra. Eine algorithmisch orientierte Einführung.) (German) Zbl 1161.68881 Berlin: Springer (ISBN 3-540-29894-0/pbk). xiii, 515 p. springer eBook: http://dx.doi.org/10.1007/3-540-29895-9 Kaplan, Michael Computer algebra. (Computeralgebra.) (German) Zbl 1093.68148 Berlin: Springer (ISBN 3-540-21379-1/pbk). xii, 391 p. springer eBook: http://dx.doi.org/10.1007/b137968

Course L0423: Algorithmic Algebra	
Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Dr. Prashant Batra
Language	DE
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1428: Linear and Nonlinear Optimization

Courses

Title	Typ	Hrs/wk	CP
Linear and Nonlinear Optimization (L2062)	Lecture	4	4
Linear and Nonlinear Optimization (L2063)	Recitation Section (large)	1	2

Module Responsible	Prof. Matthias Mnich
Admission Requirements	None
Recommended Previous Knowledge	• Discrete Algebraic Structures • Mathematics I • Graph Theory and Optimization
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students can name the basic concepts in linear and non-linear optimization. They are able to explain them using appropriate examples. • Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples. • They know proof strategies and can reproduce them.
Skills	• Students can model problems in linear and non-linear optimization with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods. • Students are able to discover and verify further logical connections between the concepts studied in the course. • For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.
Personal Competence	
Social Competence	• Students are able to work together in teams. They are capable to use mathematics as a common language. • In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.
Autonomy	• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. • Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Computer Science: Specialisation III. Mathematics: Elective Compulsory Computer Science in Engineering: Specialisation III. Mathematics: Elective Compulsory

Course L2062: Linear and Nonlinear Optimization

Typ	Lecture
Hrs/wk	4
CP	4
Workload in Hours	Independent Study Time 64, Study Time in Lecture 56
Lecturer	Prof. Matthias Mnich
Language	DE/EN
Cycle	WiSe
Content	• Modelling linear programming problems • Graphical method • Algebraic background • Convexity • Polyhedral theory • Simplex method • Degeneracy and convergence • duality • interior-point methods • quadratic optimization • integer linear programming
Literature	• A. Schrijver: Combinatorial Optimization: Polyhedra and Efficiency. Springer, 2003 • B. Korte and T. Vygen: Combinatorial Optimization: Theory and Algorithms. Springer, 2018 • T. Cormen, Ch. Leiserson, R. Rivest, C. Stein: Introduction to Algorithms. MIT Press, 2013

Course L2063: Linear and Nonlinear Optimization	
Typ	Recitation Section (large)
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Matthias Mnich
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M0716: Hierarchical Algorithms

Courses

Title	Typ	Hrs/wk	CP
Hierarchical Algorithms (L0585)	Lecture	2	3
Hierarchical Algorithms (L0586)	Recitation Section (small)	2	3

Module Responsible	Prof. Sabine Le Borne
Admission Requirements	None
Recommended Previous Knowledge	• Mathematics I, II, III for Engineering students (german or english) or Analysis & Linear Algebra I + II as well as Analysis III for Technomathematicians • Programming experience in C
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students are able to • name representatives of hierarchical algorithms and list their characteristics, • explain construction techniques for hierarchical algorithms, • discuss aspects regarding the efficient implementation of hierarchical algorithms.
Skills	Students are able to • implement the hierarchical algorithms discussed in the lecture, • analyse the storage and computational complexities of the algorithms, • adapt algorithms to problem settings of various applications and thus develop problem adapted variants.
Personal Competence	
Social Competence	Students are able to • work together in heterogeneously composed teams (i.e., teams from different study programs and background knowledge), explain theoretical foundations and support each other with practical aspects regarding the implementation of algorithms.
Autonomy	Students are capable • to assess whether the supporting theoretical and practical excercises are better solved individually or in a team, • to work on complex problems over an extended period of time, • to assess their individual progess and, if necessary, to ask questions and seek help.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	20 min
Assignment for the Following Curricula	Computer Science: Specialisation III. Mathematics: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Simulation Technology: Elective Compulsory

Course L0585: Hierarchical Algorithms

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sabine Le Borne
Language	DE/EN
Cycle	WiSe
Content	• Low rank matrices • Separable expansions • Hierarchical matrix partitions • Hierarchical matrices • Formatted matrix operations • Applications • Additional topics (e.g. H2 matrices, matrix functions, tensor products)
Literature	W. Hackbusch: Hierarchische Matrizen: Algorithmen und Analysis

Course L0586: Hierarchical Algorithms	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sabine Le Borne
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1405: Randomised Algorithms and Random Graphs

Courses

Title	Typ	Hrs/wk	CP
Randomised Algorithms and Random Graphs (L2010)	Lecture	2	3
Randomised Algorithms and Random Graphs (L2011)	Recitation Section (large)	2	3

Module Responsible	Prof. Anusch Taraz
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students can describe basic concepts in the area of Randomized Algorithms and Random Graphs such as random walks, tail bounds, fingerprinting and algebraic techniques, first and second moment methods, and various random graph models. They are able to explain them using appropriate examples. • Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples. • They know proof strategies and can apply them.
Skills	• Students can model problems with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods. • Students are able to explore and verify further logical connections between the concepts studied in the course. • For a given problem, the students can develop and execute a suitable technique, and are able to critically evaluate the results.
Personal Competence	
Social Competence	• Students are able to work together in teams. They are capable to establish a common language. • In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.
Autonomy	• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. • Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation III. Mathematics: Elective Compulsory Computer Science in Engineering: Specialisation III. Mathematics: Elective Compulsory

Course L2010: Randomised Algorithms and Random Graphs	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Anusch Taraz, Prof. Volker Turau
Language	DE/EN
Cycle	SoSe
Content	Randomized Algorithms: • introduction and recalling basic tools from probability • randomized search • random walks • text search with fingerprinting • parallel and distributed algorithms • online algorithms Random Graphs: • typical properties • first and second moment method • tail bounds • thresholds and phase transitions • probabilistic method • models for complex networks
Literature	• Motwani, Raghavan: Randomized Algorithms • Worsch: Randomisierte Algorithmen • Dietzfelbinger: Randomisierte Algorithmen • Bollobas: Random Graphs • Alon, Spencer: The Probabilistic Method • Frieze, Karonski: Random Graphs • van der Hofstad: Random Graphs and Complex Networks

Course L2011: Randomised Algorithms and Random Graphs	
Typ	Recitation Section (large)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Anusch Taraz, Prof. Volker Turau
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M0714: Numerical Methods for Ordinary Differential Equations

Courses

Title	Typ	Hrs/wk	CP
Numerical Treatment of Ordinary Differential Equations (L0576)	Lecture	2	3
Numerical Treatment of Ordinary Differential Equations (L0582)	Recitation Section (small)	2	3

Module Responsible	Prof. Daniel Ruprecht
Admission Requirements	None
Recommended Previous Knowledge	• Mathematik I, II, III für Ingenieurstudierende (deutsch oder englisch) oder Analysis & Lineare Algebra I + II sowie Analysis III für Technomathematiker • Basic knowledge of MATLAB, Python or a similar programming language
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students are able to • list numerical methods for the solution of ordinary differential equations and explain their core ideas, • formulate convergence statements for the treated numerical methods (including the assumptions about the underlying problem), • explain aspects regarding the practical realisation of a method. • select the appropriate numerical method for concrete problems, implement the numerical algorithms efficiently and interpret the numerical results
Skills	Students are able to • implement, apply and compare numerical methods for the solution of ordinary differential equations, • justify the convergence behaviour of numerical methods with respect to the posed problem and selected algorithm, • develop a suitable solution approach for a given problem, if necessary by combining of several algorithms, and to realise this approach and critically evaluate the results.
Personal Competence	
Social Competence	Students are able to • work together in heterogeneously composed teams (i.e., teams from different study programs and background knowledge), explain theoretical foundations and support each other with practical aspects regarding the implementation of algorithms.
Autonomy	Students are capable • to assess whether the supporting theoretical and practical excercises are better solved individually or in a team, • to assess their individual progress and, if necessary, to ask questions and seek help.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Bioprocess Engineering: Specialisation A - General Bioprocess Engineering: Elective Compulsory Chemical and Bioprocess Engineering: Specialisation Chemical Process Engineering: Elective Compulsory Chemical and Bioprocess Engineering: Specialisation General Process Engineering: Elective Compulsory Computer Science: Specialisation III. Mathematics: Elective Compulsory Electrical Engineering: Specialisation Control and Power Systems Engineering: Elective Compulsory Energy Systems: Core Qualification: Elective Compulsory Aircraft Systems Engineering: Core Qualification: Elective Compulsory Interdisciplinary Mathematics: Specialisation II. Numerical - Modelling Training: Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Core Qualification: Compulsory Process Engineering: Specialisation Chemical Process Engineering: Elective Compulsory Process Engineering: Specialisation Process Engineering: Elective Compulsory

Course L0576: Numerical Treatment of Ordinary Differential Equations	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Daniel Ruprecht
Language	DE/EN
Cycle	SoSe
Content	Numerical methods for Initial Value Problems • single step methods • multistep methods • stiff problems • differential algebraic equations (DAE) of index 1 Numerical methods for Boundary Value Problems • multiple shooting method • difference methods
Literature	• E. Hairer, S. Noersett, G. Wanner: Solving Ordinary Differential Equations I: Nonstiff Problems. • E. Hairer, G. Wanner: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. • D. Griffiths, D. Higham: Numerical Methods for Ordinary Differential Equations.

Course L0582: Numerical Treatment of Ordinary Differential Equations	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Daniel Ruprecht
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M1668: Probability Theory

Courses

Title	Typ	Hrs/wk	CP
Probability Theory (L2643)	Lecture	3	4
Probability Theory (L2644)	Recitation Section (small)	1	2

Module Responsible	Prof. Matthias Schulte
Admission Requirements	None
Recommended Previous Knowledge	Familiarity with the basic concepts of probability
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students can name the basic concepts in probability theory. They are able to explain them using appropriate examples. • Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples. • They know proof strategies and can reproduce them.
Skills	• Students can model problems from probability theory with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods. • Students are able to explore and verify further logical connections between the concepts studied in the course. • For a given problem, the students can develop and execute a suitable technique, and are able to critically evaluate the results.
Personal Competence	
Social Competence	• Students are able to work together (e.g. on their regular home work) and to present their results appropriately (e.g. during exercise class). • In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.
Autonomy	• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. • Students can put their knowledge in relation to the contents of other lectures. • Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation III. Mathematics: Elective Compulsory Interdisciplinary Mathematics: Specialisation II. Numerical - Modelling Training: Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory

Course L2643: Probability Theory	
Typ	Lecture
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Matthias Schulte
Language	EN
Cycle	SoSe
Content	• Measure and probability spaces • Integration and expectation • Types of stochastic convergence • Law of large numbers • Central limit theorem • Radon-Nikodym theorem • Conditional expectation • Martingales • Markov chains • Poisson processes
Literature	H. Bauer, Probability theory and elements of measure theory, second edition, Academic Press, 1981. A. Klenke, Probability Theory: A Comprehensive Course, second edition, Springer, 2014. G. F. Lawler, Introduction to Stochastic Processes, second edition, Chapman & Hall/CRC, 2006. A. N. Shiryaev, Probability, second edition, Springer, 1996.

Course L2644: Probability Theory	
Typ	Recitation Section (small)
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Matthias Schulte
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M0711: Numerical Mathematics II

Courses

Title	Typ	Hrs/wk	CP
Numerical Mathematics II (L0568)	Lecture	2	3
Numerical Mathematics II (L0569)	Recitation Section (small)	2	3

Module Responsible	Prof. Sabine Le Borne
Admission Requirements	None
Recommended Previous Knowledge	• Numerical Mathematics I • Python knowledge
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students are able to • name advanced numerical methods for interpolation, approximation, integration, eigenvalue problems, eigenvalue problems, nonlinear root finding problems and explain their core ideas, • repeat convergence statements for the numerical methods, sketch convergence proofs, • explain practical aspects of numerical methods concerning runtime and storage needs • explain aspects regarding the practical implementation of numerical methods with respect to computational and storage complexity.
Skills	Students are able to • implement, apply and compare advanced numerical methods in Python, • justify the convergence behaviour of numerical methods with respect to the problem and solution algorithm and to transfer it to related problems, • for a given problem, develop a suitable solution approach, if necessary through composition of several algorithms, to execute this approach and to critically evaluate the results
Personal Competence	
Social Competence	Students are able to • work together in heterogeneously composed teams (i.e., teams from different study programs and background knowledge), explain theoretical foundations and support each other with practical aspects regarding the implementation of algorithms.
Autonomy	Students are capable • to assess whether the supporting theoretical and practical excercises are better solved individually or in a team, • to assess their individual progess and, if necessary, to ask questions and seek help.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	25 min
Assignment for the Following Curricula	Computer Science: Specialisation III. Mathematics: Elective Compulsory Computer Science in Engineering: Specialisation III. Mathematics: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Core Qualification: Elective Compulsory

Course L0568: Numerical Mathematics II

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sabine Le Borne, Dr. Jens-Peter Zemke
Language	DE/EN
Cycle	SoSe
Content	1. Error and stability: Notions and estimates 2. Rational interpolation and approximation 3. Multidimensional interpolation (RBF) and approximation (neural nets) 4. Quadrature: Gaussian quadrature, orthogonal polynomials 5. Linear systems: Perturbation theory of decompositions, structured matrices 6. Eigenvalue problems: LR-, QD-, QR-Algorithmus 7. Nonlinear systems of equations: Newton and Quasi-Newton methods, line search (optional) 8. Krylov space methods: Arnoldi-, Lanczos methods (optional)
Literature	• Skript • Stoer/Bulirsch: Numerische Mathematik 1, Springer • Dahmen, Reusken: Numerik für Ingenieure und Naturwissenschaftler, Springer

Course L0569: Numerical Mathematics II	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sabine Le Borne, Dr. Jens-Peter Zemke
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M0881: Mathematical Image Processing

Courses

Title	Typ	Hrs/wk	CP
Mathematical Image Processing (L0991)	Lecture	3	4
Mathematical Image Processing (L0992)	Recitation Section (small)	1	2

Module Responsible	Prof. Marko Lindner
Admission Requirements	None
Recommended Previous Knowledge	• Analysis: partial derivatives, gradient, directional derivative • Linear Algebra: eigenvalues, least squares solution of a linear system
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students are able to • characterize and compare diffusion equations • explain elementary methods of image processing • explain methods of image segmentation and registration • sketch and interrelate basic concepts of functional analysis
Skills	Students are able to • implement and apply elementary methods of image processing • explain and apply modern methods of image processing
Personal Competence	
Social Competence	Students are able to work together in heterogeneously composed teams (i.e., teams from different study programs and background knowledge) and to explain theoretical foundations.
Autonomy	• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. • Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	20 min
Assignment for the Following Curricula	Bioprocess Engineering: Specialisation A - General Bioprocess Engineering: Elective Compulsory Computer Science: Specialisation III. Mathematics: Elective Compulsory Computer Science in Engineering: Specialisation III. Mathematics: Elective Compulsory Interdisciplinary Mathematics: Specialisation Computational Methods in Biomedical Imaging: Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Mechatronics: Specialisation System Design: Elective Compulsory Mechatronics: Core Qualification: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory Process Engineering: Specialisation Process Engineering: Elective Compulsory

Course L0991: Mathematical Image Processing

Typ	Lecture
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Marko Lindner
Language	DE/EN
Cycle	WiSe
Content	• basic methods of image processing • smoothing filters • the diffusion / heat equation • variational formulations in image processing • edge detection • de-convolution • inpainting • image segmentation • image registration
Literature	Bredies/Lorenz: Mathematische Bildverarbeitung

Course L0992: Mathematical Image Processing	
Typ	Recitation Section (small)
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Marko Lindner
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1552: Advanced Machine Learning

Courses

Title	Typ	Hrs/wk	CP
Advanced Machine Learning (L2322)	Lecture	2	3
Advanced Machine Learning (L2323)	Recitation Section (small)	2	3

Module Responsible	Dr. Jens-Peter Zemke
Admission Requirements	None
Recommended Previous Knowledge	1. Mathematics I-III 2. Numerical Mathematics 1/ Numerics 3. Programming skills, preferably in Python
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students are able to name, state and classify state-of-the-art neural networks and their corresponding mathematical basics. They can assess the difficulties of different neural networks.
Skills	Students are able to implement, understand, and, tailored to the field of application, apply neural networks.
Personal Competence	
Social Competence	Students can • develop and document joint solutions in small teams; • form groups to further develop the ideas and transfer them to other areas of applicability; • form a team to develop, build, and advance a software library.
Autonomy	Students are able to • correctly assess the time and effort of self-defined work; • assess whether the supporting theoretical and practical excercises are better solved individually or in a team; • define test problems for testing and expanding the methods; • assess their individual progess and, if necessary, to ask questions and seek help.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Computer Science: Specialisation III. Mathematics: Elective Compulsory Data Science: Core Qualification: Compulsory Computer Science in Engineering: Specialisation III. Mathematics: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Mechatronics: Specialisation System Design: Elective Compulsory Mechatronics: Core Qualification: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory

Course L2322: Advanced Machine Learning

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Dr. Jens-Peter Zemke
Language	DE/EN
Cycle	WiSe
Content	1. Basics: analogy; layout of neural nets, universal approximation, NP-completeness 2. Feedforward nets: backpropagation, variants of Stochastic Gradients 3. Deep Learning: problems and solution strategies 4. Deep Belief Networks: energy based models, Contrastive Divergence 5. CNN: idea, layout, FFT and Winograds algorithms, implementation details 6. RNN: idea, dynamical systems, training, LSTM 7. ResNN: idea, relation to neural ODEs 8. Standard libraries: Tensorflow, Keras, PyTorch 9. Recent trends
Literature	1. Skript 2. Online-Werke: ∘ http://neuralnetworksanddeeplearning.com/ ∘ https://www.deeplearningbook.org/

Course L2323: Advanced Machine Learning	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Dr. Jens-Peter Zemke
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1020: Numerical Methods for Partial Differential Equations

Courses

Title	Typ	Hrs/wk	CP
Numerics of Partial Differential Equations (L1247)	Lecture	2	3
Numerics of Partial Differential Equations (L1248)	Recitation Section (small)	2	3

Module Responsible	Prof. Daniel Ruprecht
Admission Requirements	None
Recommended Previous Knowledge	• Mathematik I - IV (for Engineering Students) **or** Analysis & Linear Algebra I + II for Technomathematicians • Numerical mathematics 1 • Numerical methods for ordinary differential equations
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students can classify partial differential equations according to the three basic types. • They know typical numerical methods like finite differences or finite volumes. • Students know the theoretical convergence results and other important properties of these methods.
Skills	Students are capable of formulating solution strategies for given partial differential equations, can comment on theoretical properties regarding convergence and are able to implement and test these methods.
Personal Competence	
Social Competence	Students are able of working together in heterogeneous teams (i.e., teams from different study programs and background knowledge) and to explain theoretical foundations.
Autonomy	• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. • Students have developed sufficient mental stamina to work on hard problems for an extended period of time
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation III. Mathematics: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Simulation Technology: Elective Compulsory

Course L1247: Numerics of Partial Differential Equations

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Daniel Ruprecht
Language	DE/EN
Cycle	WiSe
Content	Elementary Theory and Numerics of PDEs • types of PDEs • well posed problems • finite differences • finite volumes • applications
Literature	Dale R. Durran: Numerical Methods for Fluid Dynamics. Randall J. LeVeque: Numerical Methods for Conservation Laws.

Course L1248: Numerics of Partial Differential Equations

Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Daniel Ruprecht
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M0720: Matrix Algorithms

Courses			
Title	**Typ**	**Hrs/wk**	**CP**
Matrix Algorithms (L0984)	Lecture	2	3
Matrix Algorithms (L0985)	Recitation Section (small)	2	3

Module Responsible	Dr. Jens-Peter Zemke
Admission Requirements	None
Recommended Previous Knowledge	• Mathematics I - III • Numerical Mathematics 1/ Numerics • Basic knowledge of the programming languages Matlab and C
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students are able to 1. name, state and classify state-of-the-art Krylov subspace methods for the solution of the core problems of the engineering sciences, namely, eigenvalue problems, solution of linear systems, and model reduction; 2. state approaches for the solution of matrix equations (Sylvester, Lyapunov, Riccati).
Skills	Students are capable to 1. implement and assess basic Krylov subspace methods for the solution of eigenvalue problems, linear systems, and model reduction; 2. assess methods used in modern software with respect to computing time, stability, and domain of applicability; 3. adapt the approaches learned to new, unknown types of problem.
Personal Competence	
Social Competence	Students can • develop and document joint solutions in small teams; • form groups to further develop the ideas and transfer them to other areas of applicability; • form a team to develop, build, and advance a software library.
Autonomy	Students are able to • correctly assess the time and effort of self-defined work; • assess whether the supporting theoretical and practical excercises are better solved individually or in a team; • define test problems for testing and expanding the methods; • assess their individual progess and, if necessary, to ask questions and seek help.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	25 min
Assignment for the Following Curricula	Computer Science: Specialisation III. Mathematics: Elective Compulsory Data Science: Specialisation IV. Special Focus Area: Elective Compulsory Data Science: Specialisation I. Mathematics: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Mechatronics: Specialisation System Design: Elective Compulsory Mechatronics: Core Qualification: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Simulation Technology: Elective Compulsory

Course L0984: Matrix Algorithms	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Dr. Jens-Peter Zemke
Language	DE/EN
Cycle	WiSe
Content	• Part A: Krylov Subspace Methods: ◦ Basics (derivation, basis, Ritz, OR, MR) ◦ Arnoldi-based methods (Arnoldi, GMRes) ◦ Lanczos-based methods (Lanczos, CG, BiCG, QMR, SymmLQ, PvL) ◦ Sonneveld-based methods (IDR, BiCGStab, TFQMR, IDR(s)) • Part B: Matrix Equations: ◦ Sylvester Equation ◦ Lyapunov Equation ◦ Algebraic Riccati Equation
Literature	Skript (224 Seiten) Ergänzend können die folgenden Lehrbücher herangezogen werden: 1. Saad, Yousef. Numerical methods for large eigenvalue problems: revised edition. Society for Industrial and Applied Mathematics, 2011. 2. Saad, Yousef. Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics, 2003. 3. Van der Vorst, Henk A. Iterative Krylov methods for large linear systems. No. 13. Cambridge University Press, 2003. 4. Liesen, Jörg, and Zdenek Strakos. Krylov subspace methods: principles and analysis. Oxford University Press, 2013.

Course L0985: Matrix Algorithms	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Dr. Jens-Peter Zemke
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1865: Complexity Theory

Courses			
Title	**Typ**	**Hrs/wk**	**CP**
Complexity theory (L3062)	Lecture	2	3
Complexity theory (L3063)	Recitation Section (small)	2	3

Module Responsible	Prof. Antoine Wiehe
Admission Requirements	None
Recommended Previous Knowledge	Basic knowledge in computability and formal language theory
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory** No; **Bonus** 20 %; **Form** Excercises; **Description**
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Computer Science: Specialisation III. Mathematics: Elective Compulsory

Course L3062: Complexity theory

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Antoine Wiehe
Language	EN
Cycle	WiSe
Content	Computational complexity is a field from theoretical computer science that is concerned with the study of computational problems and their organisation in various classes corresponding to the amount of resources (like time or memory) that are needed to solve the problems. This is one of the most active research fields in theoretical computer science and a number of famous open problems are directly connected to computational complexity (for example, the Millennium problem "P vs. NP" or the complexity of the graph isomorphism problem). The course will cover the core and advanced material from this discipline, such as the important complexity classes (including, but not limited to, P and NP), as well as the classical results relating these classes.
Literature	• Computational complexity: a modern approach, S. Arora and B. Barak • Computational complexity, C. H. Papadimitriou

Course L3063: Complexity theory

Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Antoine Wiehe
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Specialization IV. Subject Specific Focus

Module M1565: Technical Complementary Course I for CSMS

Courses

Title	Typ	Hrs/wk	CP

Module Responsible	Dozenten des SD E
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Depends on choice of courses
Credit points	6
Assignment for the Following Curricula	Computer Science: Specialisation IV. Subject Specific Focus: Elective Compulsory

Module M1566: Technical Complementary Course II for CSMS

Courses			
Title	**Typ**	**Hrs/wk**	**CP**

Module Responsible	Dozenten des SD E
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Depends on choice of courses
Credit points	6
Assignment for the Following Curricula	Computer Science: Specialisation IV. Subject Specific Focus: Elective Compulsory

Module M1564: Advanced Seminars Computer Science and Communication Technology

Courses			
Title	**Typ**	**Hrs/wk**	**CP**
Advanced Seminar Computer Science and Communication Technology I (L2352)	Seminar	2	3
Introductory Seminar Computer Science and Communication Technology II (L2429)	Seminar	2	3

Module Responsible	Dozenten des SD E
Admission Requirements	None
Recommended Previous Knowledge	Basic knowledge of Computer Science and Mathematics at the Master's level.
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	The students are able to • explicate a specific topic in the field of Computer Science, • describe complex issues, • present different views and evaluate in a critical way.
Skills	The students are able to • familiarize in a specific topic of Computer Science in limited time, • realize a literature survey on the specific topic and cite in a correct way, • elaborate a presentation and give a lecture to a selected audience, • sum up the presentation in 10-15 lines, • answer questions in the final discussion.
Personal Competence	
Social Competence	The students are able to • elaborate and introduce a topic for a certain audience, • discuss the topic, content and structure of the presentation with the instructor, • discuss certain aspects with the audience, and • as the lecturer listen and respond to questions from the audience.
Autonomy	The students are able to • define the task in question in an autonomous way, • develop the necessary knowledge, • use appropriate work equipment, and • guided by an instructor critically check the working status.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Presentation
Examination duration and scale	x
Assignment for the Following Curricula	Computer Science: Specialisation IV. Subject Specific Focus: Elective Compulsory Information and Communication Systems: Specialisation Communication Systems: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems: Elective Compulsory

Course L2352: Advanced Seminar Computer Science and Communication Technology I

Typ	Seminar
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Dozenten des SD E
Language	EN
Cycle	WiSe/SoSe
Content	- Seminar presentations by enrolled students about selected topics of computer science and communication technology - Active participation in discussions
Literature	Wird vom Veranstalter bekanntgegeben.

Course L2429: Introductory Seminar Computer Science and Communication Technology II	
Typ	Seminar
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Dozenten des SD E
Language	DE/EN
Cycle	WiSe/SoSe
Content	
Literature	

Thesis

Module M-002: Master Thesis

Courses			
Title	Typ	Hrs/wk	CP

Module Responsible	Professoren der TUHH
Admission Requirements	• According to General Regulations §21 (1): At least 60 credit points have to be achieved in study programme. The examinations board decides on exceptions.
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• The students can use specialized knowledge (facts, theories, and methods) of their subject competently on specialized issues. • The students can explain in depth the relevant approaches and terminologies in one or more areas of their subject, describing current developments and taking up a critical position on them. • The students can place a research task in their subject area in its context and describe and critically assess the state of research.
Skills	The students are able: • To select, apply and, if necessary, develop further methods that are suitable for solving the specialized problem in question. • To apply knowledge they have acquired and methods they have learnt in the course of their studies to complex and/or incompletely defined problems in a solution-oriented way. • To develop new scientific findings in their subject area and subject them to a critical assessment.
Personal Competence	
Social Competence	Students can • Both in writing and orally outline a scientific issue for an expert audience accurately, understandably and in a structured way. • Deal with issues competently in an expert discussion and answer them in a manner that is appropriate to the addressees while upholding their own assessments and viewpoints convincingly.
Autonomy	Students are able: • To structure a project of their own in work packages and to work them off accordingly. • To work their way in depth into a largely unknown subject and to access the information required for them to do so. • To apply the techniques of scientific work comprehensively in research of their own.
Workload in Hours	Independent Study Time 900, Study Time in Lecture 0
Credit points	30
Course achievement	None
Examination	Thesis
Examination duration and scale	According to General Regulations
Assignment for the Following Curricula	Civil Engineering: Thesis: Compulsory Bioprocess Engineering: Thesis: Compulsory Chemical and Bioprocess Engineering: Thesis: Compulsory Computer Science: Thesis: Compulsory Data Science: Thesis: Compulsory Electrical Engineering: Thesis: Compulsory Energy Systems: Thesis: Compulsory Environmental Engineering: Thesis: Compulsory Aircraft Systems Engineering: Thesis: Compulsory Global Innovation Management: Thesis: Compulsory Computer Science in Engineering: Thesis: Compulsory Information and Communication Systems: Thesis: Compulsory Interdisciplinary Mathematics: Thesis: Compulsory International Production Management: Thesis: Compulsory International Management and Engineering: Thesis: Compulsory Joint European Master in Environmental Studies - Cities and Sustainability: Thesis: Compulsory Logistics, Infrastructure and Mobility: Thesis: Compulsory Aeronautics: Thesis: Compulsory Materials Science and Engineering: Thesis: Compulsory Materials Science: Thesis: Compulsory Mechanical Engineering and Management: Thesis: Compulsory Mechatronics: Thesis: Compulsory Biomedical Engineering: Thesis: Compulsory

	Microelectronics and Microsystems: Thesis: Compulsory Product Development, Materials and Production: Thesis: Compulsory Renewable Energies: Thesis: Compulsory Naval Architecture and Ocean Engineering: Thesis: Compulsory Ship and Offshore Technology: Thesis: Compulsory Teilstudiengang Lehramt Metalltechnik: Thesis: Compulsory Theoretical Mechanical Engineering: Thesis: Compulsory Process Engineering: Thesis: Compulsory Water and Environmental Engineering: Thesis: Compulsory Certification in Engineering & Advisory in Aviation: Thesis: Compulsory