

# **Module Manual**

Master of Science (M.Sc.)

# **Computer Science**

Cohort: Winter Term 2021

Updated: 31st May 2021

---

---

## Table of Contents

---

---

Table of Contents	2
Program description	3
Core qualification	4
Module M0523: Business & Management	4
Module M0524: Non-technical Courses for Master	5
Module M1563: Research Project Computer Science	7
Module M1564: Advanced Seminars Computer Science	8
Specialization I. Computer and Software Engineering	9
Module M0753: Software Verification	9
Module M0942: Software Security	11
Module M1694: Security of Cyber-Physical Systems	13
Module M0926: Distributed Algorithms	15
Module M1400: Design of Dependable Systems	16
Module M1685: Selected Aspects in Computer Science	18
Module M1682: Secure Software Engineering	19
Module M1397: Model Checking - Proof Engines and Algorithms	20
Module M1301: Software Testing	22
Module M1427: Algorithmic Game Theory	24
Module M1248: Compilers for Embedded Systems	26
Module M0556: Computer Graphics	28
Module M1741: Operating System Construction	30
Module M0910: Advanced System-on-Chip Design (Lab)	31
Module M0839: Traffic Engineering	32
Module M1742: Operating System Techniques	34
Specialization II: Intelligence Engineering	35
Module M0633: Industrial Process Automation	35
Module M0550: Digital Image Analysis	37
Module M0629: Intelligent Autonomous Agents and Cognitive Robotics	39
Module M0630: Robotics and Navigation in Medicine	41
Module M1702: Process Imaging	43
Module M0627: Machine Learning and Data Mining	45
Module M0623: Intelligent Systems in Medicine	47
Module M1302: Applied Humanoid Robotics	49
Module M1249: Medical Imaging	50
Specialization III. Mathematics	52
Module M0667: Algorithmic Algebra	52
Module M1428: Linear and Nonlinear Optimization	55
Module M0716: Hierarchical Algorithms	57
Module M1405: Randomised Algorithms and Random Graphs	59
Module M0714: Numerical Treatment of Ordinary Differential Equations	61
Module M1668: Probability Theory	63
Module M0711: Numerical Mathematics II	65
Module M0881: Mathematical Image Processing	67
Module M1552: Mathematics of Neural Networks	69
Module M1020: Numerics of Partial Differential Equations	71
Specialization IV. Subject Specific Focus	73
Module M1565: Technical Complementary Course I for CSMS	73
Module M1566: Technical Complementary Course II for CSMS	74
Thesis	75
Module M-002: Master Thesis	75

---

---

**Program description**

---

---

**Content**

---



---

**Core qualification**


---



---

**Module M0523: Business & Management**

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

**Courses**

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

Module M0524: Non-technical Courses for Master	
<b>Module Responsible</b>	Dagmar Richter
<b>Admission Requirements</b>	None
<b>Recommended Previous Knowledge</b>	None
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results
<b>Professional Competence</b> <i>Knowledge</i>	<p><b>The Nontechnical Academic Programms (NTA)</b></p> <p>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 <b>teaching architecture</b>, in its <b>teaching and learning arrangements</b>, in <b>teaching areas</b> and by means of teaching offerings in which students can qualify by opting for <b>specific competences</b> and a <b>competence level</b> at the Bachelor’s or Master’s level. The teaching offerings are pooled in two different catalogues for nontechnical complementary courses.</p> <p><b>The Learning Architecture</b></p> <p>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.</p> <p>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”.</p> <p>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.</p> <p><b>Teaching and Learning Arrangements</b></p> <p>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.</p> <p><b>Fields of Teaching</b></p> <p>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.</p> <p>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.</p> <p><b>The Competence Level</b></p> <p>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.</p> <p>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.</p> <p><b>Specialized Competence (Knowledge)</b></p> <p>Students can</p> <ul style="list-style-type: none"> <li>• explain specialized areas in context of the relevant non-technical disciplines,</li> <li>• outline basic theories, categories, terminology, models, concepts or artistic techniques in the disciplines represented in the learning area,</li> <li>• different specialist disciplines relate to their own discipline and differentiate it as well as make connections,</li> <li>• 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,</li> <li>• Can communicate in a foreign language in a manner appropriate to the subject.</li> </ul>
<b>Personal Competence</b> <i>Social Competence</i>	<p><b>Professional Competence (Skills)</b></p> <p>In selected sub-areas students can</p> <ul style="list-style-type: none"> <li>• apply basic and specific methods of the said scientific disciplines,</li> <li>• question a specific technical phenomena, models, theories from the viewpoint of another, aforementioned specialist discipline,</li> <li>• to handle simple and advanced questions in aforementioned scientific disciplines in a successful manner,</li> <li>• justify their decisions on forms of organization and application in practical questions in contexts that go beyond the technical relationship to the subject.</li> </ul> <p><b>Personal Competences (Social Skills)</b></p>

<p><i>Autonomy</i></p>	<p>Students will be able</p> <ul style="list-style-type: none"> <li>• to learn to collaborate in different manner,</li> <li>• to present and analyze problems in the abovementioned fields in a partner or group situation in a manner appropriate to the addressees,</li> <li>• 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),</li> <li>• to explain nontechnical items to auditorium with technical background knowledge.</li> </ul> <p><b>Personal Competences (Self-reliance)</b></p> <p>Students are able in selected areas</p> <ul style="list-style-type: none"> <li>• to reflect on their own profession and professionalism in the context of real-life fields of application</li> <li>• to organize themselves and their own learning processes</li> <li>• to reflect and decide questions in front of a broad education background</li> <li>• to communicate a nontechnical item in a competent way in written form or verbally</li> <li>• to organize themselves as an entrepreneurial subject country (as far as this study-focus would be chosen)</li> </ul>
<p><b>Workload in Hours</b></p>	<p>Depends on choice of courses</p>
<p><b>Credit points</b></p>	<p>6</p>

**Courses**

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

Module M1563: Research Project Computer Science			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Research Project Computer Science (L2353)		Projection Course	8
<b>CP</b>			12
<b>Module Responsible</b>	Prof. Karl-Heinz Zimmermann		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Basic knowledge and techniques from the Master courses in the semesters 1 and 2.		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	Students are able to acquire advanced knowledge in a subfield of Computer Science and can independently acquire deeper knowledge in the field.		
<i>Skills</i>	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.		
<b>Personal Competence</b>			
<i>Social Competence</i>	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.		
<i>Autonomy</i>	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.		
<b>Workload in Hours</b>	Independent Study Time 248, Study Time in Lecture 112		
<b>Credit points</b>	12		
<b>Course achievement</b>	None		
<b>Examination</b>	Study work		
<b>Examination duration and scale</b>	Vortrag		
<b>Assignment for the Following Curricula</b>	Computer Science: Core qualification: Compulsory		

Course L2353: Research Project Computer Science	
<b>Typ</b>	Projection Course
<b>Hrs/wk</b>	8
<b>CP</b>	12
<b>Workload in Hours</b>	Independent Study Time 248, Study Time in Lecture 112
<b>Lecturer</b>	Prof. Karl-Heinz Zimmermann
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	
<b>Literature</b>	

Module M1564: Advanced Seminars Computer Science				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Advanced Seminar Computer Science I (L2352)		Seminar	2	3
Introductory Seminar Computer Science II (L2429)		Seminar	2	3
<b>Module Responsible</b>	Prof. Karl-Heinz Zimmermann			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Basic knowledge of Computer Science and Mathematics at the Master's level.			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>				
<i>Knowledge</i>	The students are able to			
	<ul style="list-style-type: none"> <li>• explicate a specific topic in the field of Computer Science,</li> <li>• describe complex issues,</li> <li>• present different views and evaluate in a critical way.</li> </ul>			
<i>Skills</i>	The students are able to			
	<ul style="list-style-type: none"> <li>• familiarize in a specific topic of Computer Science in limited time,</li> <li>• realize a literature survey on the specific topic and cite in a correct way,</li> <li>• elaborate a presentation and give a lecture to a selected audience,</li> <li>• sum up the presentation in 10-15 lines,</li> <li>• answer questions in the final discussion.</li> </ul>			
<b>Personal Competence</b>				
<i>Social Competence</i>	The students are able to			
	<ul style="list-style-type: none"> <li>• elaborate and introduce a topic for a certain audience,</li> <li>• discuss the topic, content and structure of the presentation with the instructor,</li> <li>• discuss certain aspects with the audience, and</li> <li>• as the lecturer listen and respond to questions from the audience.</li> </ul>			
<i>Autonomy</i>	The students are able to			
	<ul style="list-style-type: none"> <li>• define the task in question in an autonomous way,</li> <li>• develop the necessary knowledge,</li> <li>• use appropriate work equipment, and</li> <li>• guided by an instructor critically check the working status.</li> </ul>			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56			
<b>Credit points</b>	6			
<b>Course achievement</b>	None			
<b>Examination</b>	Presentation			
<b>Examination duration and scale</b>	x			
<b>Assignment for the Following Curricula</b>	Computer Science: Core qualification: Elective Compulsory			

Course L2352: Advanced Seminar Computer Science I	
<b>Typ</b>	Seminar
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Karl-Heinz Zimmermann
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	
<b>Literature</b>	

Course L2429: Introductory Seminar Computer Science II	
<b>Typ</b>	Seminar
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Karl-Heinz Zimmermann
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	
<b>Literature</b>	



## 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

<b>Module Responsible</b>	Prof. Sibylle Schupp								
<b>Admission Requirements</b>	None								
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Automata theory and formal languages</li> <li>• Computational logic</li> <li>• Object-oriented programming, algorithms, and data structures</li> <li>• Functional programming or procedural programming</li> <li>• Concurrency</li> </ul>								
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results								
<b>Professional Competence</b> <i>Knowledge</i>  <i>Skills</i>  <b>Personal Competence</b> <i>Social Competence</i>  <i>Autonomy</i>	<p>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.</p> <p>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.</p> <p>Students discuss relevant topics in class. They defend their solutions orally. They communicate in English.</p> <p>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.</p>								
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56								
<b>Credit points</b>	6								
<b>Course achievement</b>	<table border="1"> <thead> <tr> <th>Compulsory</th> <th>Bonus</th> <th>Form</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Yes</td> <td>15 %</td> <td>Excercises</td> <td></td> </tr> </tbody> </table>	Compulsory	Bonus	Form	Description	Yes	15 %	Excercises	
Compulsory	Bonus	Form	Description						
Yes	15 %	Excercises							
<b>Examination</b>	Written exam								
<b>Examination duration and scale</b>	90 min								
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computational Science and 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: Compulsory International Management and Engineering: Specialisation II. Information Technology: Elective Compulsory								

Course L0629: Software Verification	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Syntax and semantics of logic-based systems</li> <li>• Deductive verification                             <ul style="list-style-type: none"> <li>◦ Specification</li> <li>◦ Proof obligations</li> <li>◦ Program properties</li> <li>◦ Automated vs. interactive theorem proving</li> </ul> </li> <li>• Model checking                             <ul style="list-style-type: none"> <li>◦ Foundations</li> <li>◦ Property languages</li> <li>◦ Tool support</li> </ul> </li> <li>• Timed automata</li> <li>• Recent developments of verification techniques and applications</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• C. Baier and J-P. Katoen, Principles of Model Checking, MIT Press 2007.</li> <li>• M. Huth and M. Bryan, Logic in Computer Science. Modelling and Reasoning about Systems, 2nd Edition, 2004.</li> <li>• Selected Research Papers</li> </ul>

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

Module M0942: Software Security			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Software Security (L1103)		Lecture	2
Software Security (L1104)		Recitation Section (small)	2
			<b>CP</b>
			3
<b>Module Responsible</b>	Prof. Riccardo Scandariato		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Familiarity with C/C++, web programming		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	Students can		
<i>Knowledge</i>	<ul style="list-style-type: none"> <li>name the main causes for security vulnerabilities in software</li> <li>explain current methods for identifying and avoiding security vulnerabilities</li> <li>explain the fundamental concepts of code-based access control</li> </ul>		
<i>Skills</i>	Students are capable of		
	<ul style="list-style-type: none"> <li>performing a software vulnerability analysis</li> <li>developing secure code</li> </ul>		
<b>Personal Competence</b>	None		
<i>Social Competence</i>	None		
<i>Autonomy</i>	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.		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	120 minutes		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computational Science and Engineering: Specialisation I. Computer Science: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems: Elective Compulsory		

Course L1103: Software Security	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Riccardo Scandariato
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>Reliability and Software Security</li> <li>Attacks exploiting character and integer representations</li> <li>Buffer overruns</li> <li>Vulnerabilities in memory management: double free attacks</li> <li>Race conditions</li> <li>SQL injection</li> <li>Cross-site scripting and cross-site request forgery</li> <li>Testing for security; taint analysis</li> <li>Type safe languages</li> <li>Development processes for secure software</li> <li>Code-based access control</li> </ul>
<b>Literature</b>	<p>M. Howard, D. LeBlanc: Writing Secure Code, 2nd edition, Microsoft Press (2002)</p> <p>G. Hoglund, G. McGraw: Exploiting Software, Addison-Wesley (2004)</p> <p>L. Gong, G. Ellison, M. Dageforde: Inside Java 2 Platform Security, 2nd edition, Addison-Wesley (2003)</p> <p>B. LaMacchia, S. Lange, M. Lyons, R. Martin, K. T. Price: .NET Framework Security, Addison-Wesley Professional (2002)</p> <p>D. Gollmann: Computer Security, 3rd edition (2011)</p>

<b>Course L1104: Software Security</b>	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Riccardo Scandariato
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	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
<b>Module Responsible</b>	Prof. Sibylle Fröschle		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	IT security, programming skills, statistics		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	<p>The students know and can explain</p> <ul style="list-style-type: none"> <li>- the threats posed by cyber attacks to cyber-physical systems (CPS)</li> <li>- concrete attacks at a technical level, e.g. on bus systems</li> <li>- security solutions specific to CPS with their capabilities and limitations</li> <li>- examples of security architectures for CPS and the requirements they guarantee</li> <li>- standard security engineering processes for CPS</li> </ul>		
<i>Skills</i>	<p>The students are able to</p> <ul style="list-style-type: none"> <li>- identify security threats and assess the risks for a given CPS</li> <li>- apply attack toolkits to analyse a networked control system, and detect attacks beyond those taught in class</li> <li>- identify and apply security solutions suitable to the requirements</li> <li>- follow security engineering processes to develop a security architecture for a given CPS</li> <li>- recognize challenges and limitations, e.g. posed by novel types of attack</li> </ul>		
<b>Personal Competence</b>			
<i>Social Competence</i>	<p>The students are able to</p> <ul style="list-style-type: none"> <li>- expertly discuss security risks and incidents of CPS and their mitigation in a solution-oriented fashion with experts and non-experts</li> <li>- foster a security culture with respect to CPS and the corresponding critical infrastructures</li> </ul>		
<i>Autonomy</i>	<p>The students are able to</p> <ul style="list-style-type: none"> <li>- follow up and critically assess current developments in the security of CPS including relevant security incidents</li> <li>- master a new topic within the area by self-study and self-initiated interaction with experts and peers.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	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		

Course L2691: Security of Cyber-Physical Systems	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sibylle Fröschle
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<p>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:</p> <ul style="list-style-type: none"> <li>Fundamentals and motivating examples</li> <li>Networked and embedded control systems                             <ul style="list-style-type: none"> <li>Bus system level attacks</li> <li>Intruder detection systems (IDS), in particular physics-based IDS</li> <li>System security architectures, including cryptographic solutions</li> </ul> </li> <li>Adversarial machine learning attacks in the physical world</li> <li>Aspects of Location and Localization</li> <li>Wireless networks and infrastructures for critical applications                             <ul style="list-style-type: none"> <li>Communication security architectures and remaining threats</li> <li>Intruder detection systems (IDS), in particular data-centric IDS</li> <li>Resilience against multi-instance attacks</li> </ul> </li> <li>Security Engineering of CPS: Process and Norms</li> </ul>
<b>Literature</b>	Recent scientific papers and reports in the public domain.

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

Module M0926: Distributed Algorithms			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Distributed Algorithms (L1071)		Lecture	2
Distributed Algorithms (L1072)		Recitation Section (large)	2
			<b>CP</b>
			3
<b>Module Responsible</b>	Prof. Volker Turau		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Algorithms and data structures</li> <li>Distributed systems</li> <li>Discrete mathematics</li> <li>Graph theory</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> 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.</p> <p><i>Skills</i> Students design their own distributed algorithms and analyze their complexity. They make use of known standard algorithms. They compute the complexity of randomized algorithms.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p><i>Autonomy</i></p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	45 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computational Science and Engineering: Specialisation I. Computer Science: Elective Compulsory		

Course L1071: Distributed Algorithms	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Volker Turau
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>Leader Election</li> <li>Colorings &amp; Independent Sets</li> <li>Tree Algorithms</li> <li>Minimal Spanning Trees</li> <li>Randomized Distributed Algorithms</li> <li>Mutual Exclusion</li> </ul>
<b>Literature</b>	<ol style="list-style-type: none"> <li>David Peleg: Distributed Computing - A Locality-Sensitive Approach. SIAM Monograph, 2000</li> <li>Gerard Tel: Introduction to Distributed Algorithms, Cambridge University Press, 2nd edition, 2000</li> <li>Nancy Lynch: Distributed Algorithms. Morgan Kaufmann, 1996</li> <li>Volker Turau: Algorithmische Graphentheorie. Oldenbourg Wissenschaftsverlag, 3. Auflage, 2004.</li> </ol>

Course L1072: Distributed Algorithms	
<b>Typ</b>	Recitation Section (large)
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Volker Turau
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1400: Design of Dependable Systems				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Designing Dependable Systems (L2000)		Lecture	2	3
Designing Dependable Systems (L2001)		Recitation Section (small)	2	3
<b>Module Responsible</b>	Prof. Görschwin Fey			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Basic knowledge about data structures and algorithms			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <p>In the following "dependable" summarizes the concepts Reliability, Availability, Maintainability, Safety and Security.</p> <p>Knowledge about approaches for designing dependable systems, e.g.,</p> <ul style="list-style-type: none"> <li>• Structural solutions like modular redundancy</li> <li>• Algorithmic solutions like handling byzantine faults or checkpointing</li> </ul> <p>Knowledge about methods for the analysis of dependable systems</p> <p><i>Skills</i></p> <p>Ability to implement dependable systems using the above approaches.</p> <p>Ability to analyze the dependability of systems using the above methods for analysis.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p>Students</p> <ul style="list-style-type: none"> <li>• discuss relevant topics in class and</li> <li>• present their solutions orally.</li> </ul> <p><i>Autonomy</i></p> <p>Using accompanying material students independently learn in-depth relations between concepts explained in the lecture and additional solution strategies.</p>			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	Yes	None	Subject theoretical and practical work	Die Lösung einer Aufgabe ist Zulassungsvoraussetzung für die Prüfung. Die Aufgabe wird in Vorlesung und Übung definiert.
<b>Examination</b>	Oral exam			
<b>Examination duration and scale</b>	30 min			
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computational Science and 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	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Görschwin Fey
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<p>Description</p> <p>The term dependability comprises various aspects of a system. These are typically:</p> <ul style="list-style-type: none"> <li>• Reliability</li> <li>• Availability</li> <li>• Maintainability</li> <li>• Safety</li> <li>• Security</li> </ul> <p>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.</p> <p>Contents</p> <p>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:</p> <ul style="list-style-type: none"> <li>• Modelling</li> <li>• Fault Tolerance</li> <li>• Design Concepts</li> <li>• Analysis Techniques</li> </ul>
<b>Literature</b>	

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

Module M1685: Selected Aspects in Computer Science			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b> <b>CP</b>
Selected Aspects in Computer Science (L2672)		Lecture	3                  4
Selected Aspects in Computer Science (L2673)		Recitation Section (small)	1                  2
<b>Module Responsible</b>	Prof. Görschwin Fey		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b> <i>Knowledge</i> <i>Skills</i>			
<b>Personal Competence</b> <i>Social Competence</i> <i>Autonomy</i>			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	30 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory		

Course L2672: Selected Aspects in Computer Science	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 78, Study Time in Lecture 42
<b>Lecturer</b>	Dozenten des SD E
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	
<b>Literature</b>	

Course L2673: Selected Aspects in Computer Science	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	1
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 46, Study Time in Lecture 14
<b>Lecturer</b>	Dozenten des SD E
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1682: Secure Software Engineering			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Secure Software Engineering (L2667)	Lecture	2	3
Secure Software Engineering (L2668)	Project-/problem-based Learning	2	3
<b>Module Responsible</b>	Prof. Riccardo Scandariato		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b> <i>Knowledge</i> <i>Skills</i>			
<b>Personal Competence</b> <i>Social Competence</i> <i>Autonomy</i>			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	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 Software: Elective Compulsory		

Course L2667: Secure Software Engineering	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Riccardo Scandariato
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	
<b>Literature</b>	

Course L2668: Secure Software Engineering	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Riccardo Scandariato
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	
<b>Literature</b>	

Module M1397: Model Checking - Proof Engines and Algorithms				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Model Checking - Proof Engines and Algorithms (L1979)		Lecture	2	3
Model Checking - Proof Engines and Algorithms (L1980)		Recitation Section (small)	2	3
<b>Module Responsible</b>	Prof. Görschwin Fey			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Basic knowledge about data structures and algorithms			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>				
<i>Knowledge</i>	Students know			
	<ul style="list-style-type: none"> <li>algorithms and data structures for model checking,</li> <li>basics of Boolean reasoning engines and</li> <li>the impact of specification and modelling on the computational effort for model checking.</li> </ul>			
<i>Skills</i>	Students can			
	<ul style="list-style-type: none"> <li>explain and implement algorithms and data structures for model checking,</li> <li>decide whether a given problem can be solved using Boolean reasoning or model checking, and</li> <li>implement the respective algorithms.</li> </ul>			
<b>Personal Competence</b>				
<i>Social Competence</i>	Students			
	<ul style="list-style-type: none"> <li>discuss relevant topics in class and</li> <li>defend their solutions orally.</li> </ul>			
<i>Autonomy</i>	Using accompanying material students independently learn in-depth relations between concepts explained in the lecture and additional solution strategies.			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	Yes	None	Subject theoretical and practical work	Die Aufgabe wird im Rahmen von Volresung und Prüfung definiert. Die Lösung der Aufgabe ist Zulassungsvoraussetzung für die Prüfung.
<b>Examination</b>	Oral exam			
<b>Examination duration and scale</b>	30 min			
<b>Assignment for the Following Curricula</b>	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	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Görschwin Fey
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<p>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."</p> <p>And how do the underlying reasoning algorithms work so effectively in practice despite a computational complexity of NP hardness and beyond?</p> <p>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.</p> <p>Among other topics, the lecture will consider the following topics:</p> <ul style="list-style-type: none"> <li>• Modelling digital Hardware, Software, and Cyber Physical Systems</li> <li>• Data structures, decision procedures and proof engines                         <ul style="list-style-type: none"> <li>◦ Binary Decision Diagrams</li> <li>◦ And-Inverter-Graphs</li> <li>◦ Boolean Satisfiability</li> <li>◦ Satisfiability Modulo Theories</li> </ul> </li> <li>• Specification Languages                         <ul style="list-style-type: none"> <li>◦ CTL</li> <li>◦ LTL</li> <li>◦ System Verilog Assertions</li> </ul> </li> <li>• Algorithms for                         <ul style="list-style-type: none"> <li>◦ Reachability Analysis</li> <li>◦ Symbolic CTL Checking</li> <li>◦ Bounded LTL-Model Checking</li> <li>◦ Optimizations, e.g., induction, abstraction</li> </ul> </li> <li>• Quality assurance</li> </ul>
<b>Literature</b>	<p>Edmund M. Clarke, Jr., Orna Grumberg, and Doron A. Peled. 1999. <i>Model Checking</i>. MIT Press, Cambridge, MA, USA.</p> <p>A. Biere, A. Biere, M. Heule, H. van Maaren, and T. Walsh. 2009. <i>Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications</i>. IOS Press, Amsterdam, The Netherlands, The Netherlands.</p> <p>Selected research papers</p>

Course L1980: Model Checking - Proof Engines and Algorithms	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Görschwin Fey
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	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	
<b>Module Responsible</b>	Prof. Sibylle Schupp			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Software Engineering</li> <li>• Higher Programming Languages</li> <li>• Object-Oriented Programming</li> <li>• Algorithms and Data Structures</li> <li>• Experience with (Small) Software Projects</li> <li>• Statistics</li> </ul>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <p>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.</p> <p><i>Skills</i></p> <p>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.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p>Students discuss relevant topics in class. They defend their solutions orally. They communicate in English.</p> <p><i>Autonomy</i></p> <p>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</p>			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56			
<b>Credit points</b>	6			
<b>Course achievement</b>	None			
<b>Examination</b>	Subject theoretical and practical work			
<b>Examination duration and scale</b>	Software			
<b>Assignment for the Following Curricula</b>	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	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Fundamentals of software testing</li> <li>• Model-based testing</li> <li>• Test automation</li> <li>• Criteria-based testing</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• M. Pezze and M. Young, Software Testing and Analysis, John Wiley 2008.</li> <li>• P. Ammann and J. Offutt, "Introduction to Software Testing", 2nd edition 2016.</li> <li>• A. Zeller: "Why Programs Fail: A Guide to Systematic Debugging", 2nd edition 2012.</li> </ul>

Course L1792: Software Testing	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Fundamentals of software testing</li> <li>• Model-based testing</li> <li>• Test automation</li> <li>• Criteria-based testing</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• M. Pezze and M. Young, Software Testing and Analysis, John Wiley 2008.</li> <li>• P. Ammann and J. Offutt, "Introduction to Software Testing", 2nd edition 2015.</li> </ul>

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
<b>Module Responsible</b>	Prof. Matthias Mnich		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Mathematics I</li> <li>• Mathematics II</li> <li>• Algorithms and Data Structures</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b> <i>Knowledge</i>	<ul style="list-style-type: none"> <li>• Students can name the basic concepts in algorithmic game theory and mechanism design. They are able to explain them using appropriate examples.</li> <li>• Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>• They know game and mechanism design strategies and can reproduce them.</li> </ul>		
<i>Skills</i>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• Students are able to discover and verify further logical connections between the concepts studied in the course.</li> <li>• For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.</li> </ul>		
<b>Personal Competence</b> <i>Social Competence</i>	<ul style="list-style-type: none"> <li>• Students are able to work together in teams. They are capable to use mathematics as a common language.</li> <li>• 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.</li> </ul>		
<i>Autonomy</i>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computational Science and Engineering: Specialisation I. Computer Science: Elective Compulsory		



Course L2060: Algorithmic game theory	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 92, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Matthias Mnich
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<p>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.</p> <p>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.</p> <p>Topics:</p> <ul style="list-style-type: none"> <li>• basic equilibrium concepts (Nash equilibria, correlated equilibria, ...)</li> <li>• strategic actions (best-response dynamics, no-regret dynamics, ...)</li> <li>• auction design (revenue-maximizing auctions, Vickrey auctions)</li> <li>• stable matching theory (preference aggregations, kidney exchanges, ...)</li> <li>• price of anarchy and selfish routing (Braess' paradox, congestion games, ...)</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• T. Roughgarden: Twenty Lectures on Algorithmic Game Theory, Cambridge University Press, 2016.</li> <li>• N. Nisan, T. Roughgarden, E. Tardos, V. Vazirani. Algorithmic Game Theory. Cambridge University Press, 2007.</li> </ul>

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

Module M1248: Compilers for Embedded Systems			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Compilers for Embedded Systems (L1692)		Lecture	3
Compilers for Embedded Systems (L1693)		Project-/problem-based Learning	1
<b>CP</b>			
			4
			2
<b>Module Responsible</b>	Prof. Heiko Falk		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Module "Embedded Systems" C/C++ Programming skills		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <p>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</p> <ul style="list-style-type: none"> <li>to illustrate the structure and organization of such compilers,</li> <li>to distinguish and explain intermediate representations of various abstraction levels, and</li> <li>to assess optimizations and their underlying problems in all compiler phases.</li> </ul> <p>The high demands on compilers for embedded systems make effective code optimizations mandatory. The students learn in particular,</p> <ul style="list-style-type: none"> <li>which kinds of optimizations are applicable at the source code level,</li> <li>how the translation from source code to assembly code is performed,</li> <li>which kinds of optimizations are applicable at the assembly code level,</li> <li>how register allocation is performed, and</li> <li>how memory hierarchies can be exploited effectively.</li> </ul> <p>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.</p> <p><i>Skills</i></p> <p>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.</p> <p>While attending the labs, the students will learn to implement a fully functional compiler including optimizations.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p>Students are able to solve similar problems alone or in a group and to present the results accordingly.</p> <p><i>Autonomy</i></p> <p>Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.</p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	30 min		
<b>Assignment for the Following Curricula</b>	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: Technical Complementary Course: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory		

Course L1692: Compilers for Embedded Systems	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 78, Study Time in Lecture 42
<b>Lecturer</b>	Prof. Heiko Falk
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction and Motivation</li> <li>• Compilers for Embedded Systems - Requirements and Dependencies</li> <li>• Internal Structure of Compilers</li> <li>• Pre-Pass Optimizations</li> <li>• HIR Optimizations and Transformations</li> <li>• Code Generation</li> <li>• LIR Optimizations and Transformations</li> <li>• Register Allocation</li> <li>• WCET-Aware Compilation</li> <li>• Outlook</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Peter Marwedel. Embedded System Design - Embedded Systems Foundations of Cyber-Physical Systems. 2<sup>nd</sup> Edition, Springer, 2012.</li> <li>• Steven S. Muchnick. Advanced Compiler Design and Implementation. Morgan Kaufmann, 1997.</li> <li>• Andrew W. Appel. Modern compiler implementation in C. Oxford University Press, 1998.</li> </ul>

Course L1693: Compilers for Embedded Systems	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	1
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 46, Study Time in Lecture 14
<b>Lecturer</b>	Prof. Heiko Falk
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	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
<b>Module Responsible</b>	Prof. Tobias Knopp		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Linear Algebra (in particular matrix/vector computation)</li> <li>• Basic programming skills in C/C++</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> Students can explain and describe basic algorithms in 3D computer graphics.</p> <p><i>Skills</i> Students are capable of</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• apply geometric transformations (e.g. rotation, scaling) in 2D and 3D computer graphics.</li> <li>• using well-known 2D/3D APIs (OpenGL, Cairo) for solving a given problem statement.</li> </ul>		
<b>Personal Competence</b>	<p><i>Social Competence</i> Students can collaborate in a small team on the realization and validation of a 3D computer graphics pipeline.</p> <p><i>Autonomy</i></p> <ul style="list-style-type: none"> <li>• Students are able to solve simple tasks independently with reference to the contents of the lectures and the exercise sets.</li> <li>• Students are able to solve detailed problems independently with the aid of the tutorial's programming task.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Information and Communication Systems: Specialisation Communication Systems, Focus Signal Processing: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems, Focus Software and Signal Processing: Elective Compulsory International Management and Engineering: Specialisation II. Information Technology: Elective Compulsory		

Course L0145: Computer Graphics	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Tobias Knopp
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<p>Computer graphics and animation are leading to an unprecedented visual revolution. The course deals with its technological foundations:</p> <ul style="list-style-type: none"> <li>• Object-oriented Computer Graphics</li> <li>• Projections and Transformations</li> <li>• Polygonal and Parametric Modelling</li> <li>• Illuminating, Shading, Rendering</li> <li>• Computer Animation Techniques</li> <li>• Kinematics and Dynamics Effects</li> </ul> <p>Students will 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.</p>
<b>Literature</b>	<p>Alan H. Watt: 3D Computer Graphics. Harlow: Pearson (3rd ed., repr., 2009).</p> <p>Dariush Derakhshani: Introducing Autodesk Maya 2014. New York, NY : Wiley (2013).</p>

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

Module M1741: Operating System Construction				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Operating System Construction (L2812)		Lecture	2	3
Operating System Construction (L2814)		Project-/problem-based Learning	2	2
Operating System Construction (L2813)		Recitation Section (large)	1	1
<b>Module Responsible</b>	Prof. Christian Dietrich			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>				
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b> <i>Knowledge</i> <i>Skills</i>				
<b>Personal Competence</b> <i>Social Competence</i> <i>Autonomy</i>				
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	No	20 %	Subject	theoretical and practical work
<b>Examination</b>	Oral exam			
<b>Examination duration and scale</b>	25 min			
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory			

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

Course L2814: Operating System Construction	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Christian Dietrich
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	
<b>Literature</b>	

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

Module M0910: Advanced System-on-Chip Design (Lab)			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Advanced System-on-Chip Design (L1061)	Project-/problem-based Learning	3	6
<b>Module Responsible</b>	Prof. Heiko Falk		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Successful completion of the practical FPGA lab of module "Computer Architecture" is a mandatory prerequisite.		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> This module provides in-depth, hands-on experience on advanced concepts of computer architecture. Using the Hardware Description Language VHDL and using reconfigurable FPGA hardware boards, students learn how to design complex computer systems (so-called systems-on-chip, SoCs), that are commonly found in the domain of embedded systems, in actual hardware.</p> <p>Starting with a simple processor architecture, the students learn to how realize instruction-processing of a computer processor according to the principle of pipelining. They implement different styles of cache-based memory hierarchies, examine strategies for dynamic scheduling of machine instructions and for branch prediction, and finally construct a complex MPSoC system (multi-processor system-on-chip) that consists of multiple processor cores that are connected via a shared bus.</p> <p><i>Skills</i> Students will be able to analyze, how highly specific and individual computer systems can be constructed using a library of given standard components. They evaluate the interferences between the physical structure of a computer system and the software executed thereon. This way, they will be enabled to estimate the effects of design decision at the hardware level on the performance of the entire system, to evaluate the whole and complex system and to propose design options to improve a system.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> Students are able to solve similar problems alone or in a group and to present the results accordingly.</p> <p><i>Autonomy</i> Students are able to acquire new knowledge from specific literature, to transform this knowledge into actual implementations of complex hardware structures, and to associate this knowledge with contents of other classes.</p>		
<b>Workload in Hours</b>	Independent Study Time 138, Study Time in Lecture 42		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Subject theoretical and practical work		
<b>Examination duration and scale</b>	VHDL Codes and FPGA-based implementations		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory		
Course L1061: Advanced System-on-Chip Design			
<b>Typ</b>	Project-/problem-based Learning		
<b>Hrs/wk</b>	3		
<b>CP</b>	6		
<b>Workload in Hours</b>	Independent Study Time 138, Study Time in Lecture 42		
<b>Lecturer</b>	Prof. Heiko Falk		
<b>Language</b>	DE/EN		
<b>Cycle</b>	WiSe		
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction into fundamental technologies (FPGAs, MIPS single-cycle machine)</li> <li>• Pipelined instruction execution</li> <li>• Cache-based memory hierarchies</li> <li>• Busses and their arbitration</li> <li>• Multi-Processor Systems-on-Chip</li> <li>• Optional: Advanced pipelining concepts (dynamic scheduling, branch prediction)</li> </ul>		
<b>Literature</b>	<ul style="list-style-type: none"> <li>• D. Patterson, J. Hennessy. Rechnerorganisation und -entwurf. Elsevier, 2005.</li> <li>• A. Tanenbaum, J. Goodman. Computerarchitektur. Pearson, 2001.</li> <li>• A. Clements. The Principles of Computer Hardware. 3. Auflage, Oxford University Press, 2000.</li> </ul>		

Module M0839: Traffic Engineering	
<b>Courses</b>	
<b>Title</b>	<b>Typ</b> <b>Hrs/wk</b> <b>CP</b>
Seminar Traffic Engineering (L0902)	Seminar 2 2
Traffic Engineering (L0900)	Lecture 2 2
Traffic Engineering Exercises (L0901)	Recitation Section (small) 1 2
<b>Module Responsible</b>	Prof. Andreas Timm-Giel
<b>Admission Requirements</b>	None
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Fundamentals of communication or computer networks</li> <li>Stochastics</li> </ul>
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results
<b>Professional Competence</b>	
<i>Knowledge</i>	Students are able to describe methods for planning, optimisation and performance evaluation of communication networks.
<i>Skills</i>	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.
<b>Personal Competence</b>	
<i>Social Competence</i>	
<i>Autonomy</i>	Students are able to acquire the necessary expert knowledge to understand the functionality and performance of new communication networks independently.
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70
<b>Credit points</b>	6
<b>Course achievement</b>	None
<b>Examination</b>	Oral exam
<b>Examination duration and scale</b>	30 min
<b>Assignment for the Following Curricula</b>	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	
<b>Typ</b>	Seminar
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Andreas Timm-Giel, Dr. Phuong Nga Tran
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	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.
<b>Literature</b>	<ul style="list-style-type: none"> <li>U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Vieweg + Teubner</li> <li>further literature announced in the lecture</li> </ul>



Course L0900: Traffic Engineering	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Andreas Timm-Giel, Dr. Phuong Nga Tran
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<p>Network Planning and Optimization</p> <ul style="list-style-type: none"> <li>• Linear Programming (LP)</li> <li>• Network planning with LP solvers</li> <li>• Planning of communication networks</li> </ul> <p>Queueing Theory for Communication Networks</p> <ul style="list-style-type: none"> <li>• Stochastic processes</li> <li>• Queueing systems</li> <li>• Switches (circuit- and packet switching)</li> <li>• Network of queues</li> </ul>
<b>Literature</b>	<p>Literatur:</p> <p>U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Springer</p> <p>Weitere Literatur wird in der Lehrveranstaltung bekanntgegeben</p> <p>/</p> <p>Literature:</p> <p>U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Springer</p> <p>further literature announced in the lecture</p>

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

Module M1742: Operating System Techniques			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Operating System Techniques (L2815)	Lecture	1	2
Operating System Techniques (L2816)	Project-/problem-based Learning	3	4
<b>Module Responsible</b>	Prof. Christian Dietrich		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b> <i>Knowledge</i> <i>Skills</i>			
<b>Personal Competence</b> <i>Social Competence</i> <i>Autonomy</i>			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Subject theoretical and practical work		
<b>Examination duration and scale</b>	40 minutes, contents of course and practical labs		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory		

Course L2815: Operating System Techniques	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	1
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 46, Study Time in Lecture 14
<b>Lecturer</b>	Prof. Christian Dietrich
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	
<b>Literature</b>	

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

## 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
<b>Module Responsible</b>	Prof. Alexander Schlaefer		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	mathematics and optimization methods principles of automata principles of algorithms and data structures programming skills		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> 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'.</p> <p><i>Skills</i> 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.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> The students work in teams to solve problems.</p> <p><i>Autonomy</i> The students can reflect their knowledge and document the results of their work.</p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>
	No	10 %	Exercises
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 minutes		
<b>Assignment for the Following Curricula</b>	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 Aircraft Systems Engineering: Specialisation Cabin Systems: 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: Technical Complementary Course: 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	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Alexander Schlaefer
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>- foundations of problem solving and system modeling, discrete event systems</li> <li>- properties of processes, modeling using automata and Petri-nets</li> <li>- design considerations for processes (mutex, deadlock avoidance, liveness)</li> <li>- optimal scheduling for processes</li> <li>- optimal decisions when planning manufacturing systems, decisions under uncertainty</li> <li>- software design and software architectures for automation, PLCs</li> </ul>
<b>Literature</b>	<p>J. Lunze: „Automatisierungstechnik“, Oldenbourg Verlag, 2012</p> <p>Reisig: Petrinetze: Modellierungstechnik, Analysemethoden, Fallstudien; Vieweg+Teubner 2010</p> <p>Hrúz, Zhou: Modeling and Control of Discrete-event Dynamic Systems; Springer 2007</p> <p>Li, Zhou: Deadlock Resolution in Automated Manufacturing Systems, Springer 2009</p> <p>Pinedo: Planning and Scheduling in Manufacturing and Services, Springer 2009</p>

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

Module M0550: Digital Image Analysis			
Courses			
Title	Typ	Hrs/wk	CP
Digital Image Analysis (L0126)	Lecture	4	6
<b>Module Responsible</b>	Prof. Rolf-Rainer Grigat		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	System theory of one-dimensional signals (convolution and correlation, sampling theory, interpolation and decimation, Fourier transform, linear time-invariant systems), linear algebra (Eigenvalue decomposition, SVD), basic stochastics and statistics (expectation values, influence of sample size, correlation and covariance, normal distribution and its parameters), basics of Matlab, basics in optics		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> Students can</p> <ul style="list-style-type: none"> <li>• Describe imaging processes</li> <li>• Depict the physics of sensorics</li> <li>• Explain linear and non-linear filtering of signals</li> <li>• Establish interdisciplinary connections in the subject area and arrange them in their context</li> <li>• Interpret effects of the most important classes of imaging sensors and displays using mathematical methods and physical models.</li> </ul> <p><i>Skills</i> Students are able to</p> <ul style="list-style-type: none"> <li>• Use highly sophisticated methods and procedures of the subject area</li> <li>• Identify problems and develop and implement creative solutions.</li> </ul> <p>Students can solve simple arithmetical problems relating to the specification and design of image processing and image analysis systems.</p> <p>Students are able to assess different solution approaches in multidimensional decision-making areas.</p> <p>Students can undertake a prototypical analysis of processes in Matlab.</p>		
<b>Personal Competence</b>	<p><i>Social Competence</i> k.A.</p> <p><i>Autonomy</i> Students can solve image analysis tasks independently using the relevant literature.</p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	60 Minutes, Content of Lecture and materials in StudIP		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation II: Intelligence Engineering: Elective Compulsory Electrical Engineering: Specialisation Information and Communication Systems: Elective Compulsory Electrical Engineering: Specialisation Medical Technology: Elective Compulsory Information and Communication Systems: Specialisation Communication Systems, Focus Signal Processing: Elective Compulsory Information and Communication Systems: Specialisation Secure and Dependable IT Systems, Focus Software and Signal Processing: Elective Compulsory International Management and Engineering: Specialisation II. Information Technology: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Microelectronics and Microsystems: Specialisation Communication and Signal Processing: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory		

Course L0126: Digital Image Analysis	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	4
<b>CP</b>	6
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56
<b>Lecturer</b>	Prof. Rolf-Rainer Grigat
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Image representation, definition of images and volume data sets, illumination, radiometry, multispectral imaging, reflectivities, shape from shading</li> <li>• Perception of luminance and color, color spaces and transforms, color matching functions, human visual system, color appearance models</li> <li>• imaging sensors (CMOS, CCD, HDR, X-ray, IR), sensor characterization(EMVA1288), lenses and optics</li> <li>• spatio-temporal sampling (interpolation, decimation, aliasing, leakage, moiré, flicker, apertures)</li> <li>• features (filters, edge detection, morphology, invariance, statistical features, texture)</li> <li>• optical flow ( variational methods, quadratic optimization, Euler-Lagrange equations)</li> <li>• segmentation (distance, region growing, cluster analysis, active contours, level sets, energy minimization and graph cuts)</li> <li>• registration (distance and similarity, variational calculus, iterative closest points)</li> </ul>
<b>Literature</b>	Bredies/Lorenz, Mathematische Bildverarbeitung, Vieweg, 2011 Wedel/Cremers, Stereo Scene Flow for 3D Motion Analysis, Springer 2011 Handels, Medizinische Bildverarbeitung, Vieweg, 2000 Pratt, Digital Image Processing, Wiley, 2001 Jain, Fundamentals of Digital Image Processing, Prentice Hall, 1989

Module M0629: Intelligent Autonomous Agents and Cognitive Robotics			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Intelligent Autonomous Agents and Cognitive Robotics (L0341)		Lecture	2
Intelligent Autonomous Agents and Cognitive Robotics (L0512)		Recitation Section (small)	2
<b>Module Responsible</b>	Rainer Marrone		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Vectors, matrices, Calculus		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	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.		
<i>Skills</i>	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.		
<b>Personal Competence</b>			
<i>Social Competence</i>	Students are able to discuss their solutions to problems with others. They communicate in English		
<i>Autonomy</i>	Students are able of checking their understanding of complex concepts by solving variants of concrete problems		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 minutes		
<b>Assignment for the Following Curricula</b>	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: Technical Complementary Course: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory		

Course L0341: Intelligent Autonomous Agents and Cognitive Robotics	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 92, Study Time in Lecture 28
<b>Lecturer</b>	Rainer Marrone
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Definition of agents, rational behavior, goals, utilities, environment types</li> <li>• Adversarial agent cooperation: Agents with complete access to the state(s) of the environment, games, Minimax algorithm, alpha-beta pruning, elements of chance</li> <li>• 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</li> <li>• 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).</li> <li>• 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</li> <li>• Decision making under uncertainty: Simple decisions: utility theory, multivariate utility functions, dominance, decision networks, value of information Complex decisions: sequential decision problems, value iteration, policy iteration, MDPs Decision-theoretic agents: POMDPs, reduction to multidimensional continuous MDPs, dynamic decision networks</li> <li>• Simultaneous Localization and Mapping</li> <li>• Planning</li> <li>• Game theory (Golden Balls: Split or Share) Decisions with multiple agents, Nash equilibrium, Bayes-Nash equilibrium</li> <li>• Social Choice Voting protocols, preferences, paradoxes, Arrow's Theorem,</li> <li>• 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</li> </ul>
<b>Literature</b>	<ol style="list-style-type: none"> <li>1. Artificial Intelligence: A Modern Approach (Third Edition), Stuart Russell, Peter Norvig, Prentice Hall, 2010, Chapters 2-5, 10-11, 13-17</li> <li>2. Probabilistic Robotics, Thrun, S., Burgard, W., Fox, D. MIT Press 2005</li> <li>3. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Yoav Shoham, Kevin Leyton-Brown, Cambridge University Press, 2009</li> </ol>

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



Module M0630: Robotics and Navigation in Medicine				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
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
<b>Module Responsible</b>	Prof. Alexander Schlaefer			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>principles of math (algebra, analysis/calculus)</li> <li>principles of programming, e.g., in Java or C++</li> <li>solid R or Matlab skills</li> </ul>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>				
<i>Knowledge</i>	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.			
<i>Skills</i>	The students are able to design and evaluate navigation systems and robotic systems for medical applications.			
<b>Personal Competence</b>				
<i>Social Competence</i>	The students discuss the results of other groups, provide helpful feedback and can incorporate feedback into their work.			
<i>Autonomy</i>	The students can reflect their knowledge and document the results of their work. They can present the results in an appropriate manner.			
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	Yes	10 %	Written elaboration	
	Yes	10 %	Presentation	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 minutes			
<b>Assignment for the Following Curricula</b>	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: Technical Complementary Course: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Bio- and Medical Technology: Elective Compulsory			

Course L0335: Robotics and Navigation in Medicine	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Alexander Schlaefer
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>kinematics</li> <li>calibration</li> <li>tracking systems</li> <li>navigation and image guidance</li> <li>motion compensation</li> </ul> The seminar extends and complements the contents of the lecture with respect to recent research results.
<b>Literature</b>	Spong et al.: Robot Modeling and Control, 2005 Troccaz: Medical Robotics, 2012 Further literature will be given in the lecture.

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

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

Module M1702: Process Imaging			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Process Imaging (L2723)		Lecture	2
Process Imaging (L2724)		Project-/problem-based Learning	2
			<b>CP</b>
			3
<b>Module Responsible</b>	Prof. Alexander Penn		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b> <i>Knowledge</i> <i>Skills</i>			
<b>Personal Competence</b> <i>Social Competence</i> <i>Autonomy</i>			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	Bioprocess Engineering: Specialisation A - General Bioprocess Engineering: Elective Compulsory Bioprocess Engineering: Specialisation A - General Bioprocess Engineering: Elective Compulsory Bioprocess Engineering: Specialisation B - Industrial 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 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 General Process Engineering: Elective Compulsory Chemical and Bioprocess Engineering: Specialisation Bioprocess Engineering: Elective Compulsory Chemical and Bioprocess Engineering: Specialisation Bioprocess Engineering: Elective Compulsory Chemical and Bioprocess Engineering: Specialisation Chemical Process 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 Process Engineering: Elective Compulsory Process Engineering: Specialisation Chemical Process Engineering: Elective Compulsory Process Engineering: Specialisation Chemical Process Engineering: Elective Compulsory Process Engineering: Specialisation Environmental 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 Environment: Elective Compulsory Water and Environmental Engineering: Specialisation Water: Elective Compulsory Water and Environmental Engineering: Specialisation Water: Elective Compulsory		

Course L2723: Process Imaging	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Alexander Penn
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	
<b>Literature</b>	

Course L2724: Process Imaging	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Alexander Penn
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	
<b>Literature</b>	

Module M0627: Machine Learning and Data Mining			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Machine Learning and Data Mining (L0340)	Lecture	2	4
Machine Learning and Data Mining (L0510)	Recitation Section (small)	2	2
<b>Module Responsible</b>	NN		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Calculus</li> <li>• Stochastics</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <p>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.</p> <p><i>Skills</i></p> <p>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 learning. 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.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p><i>Autonomy</i></p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 minutes		
<b>Assignment for the Following Curricula</b>	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: Technical Complementary Course: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory		

Course L0340: Machine Learning and Data Mining	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 92, Study Time in Lecture 28
<b>Lecturer</b>	Rainer Marrone
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Decision trees</li> <li>• First-order inductive learning</li> <li>• Incremental learning: Version spaces</li> <li>• Uncertainty</li> <li>• Bayesian networks</li> <li>• Learning parameters of Bayesian networks BME, MAP, ML, EM algorithm</li> <li>• Learning structures of Bayesian networks</li> <li>• Gaussian Mixture Models</li> <li>• kNN classifier, neural network classifier, support vector machine (SVM) classifier</li> <li>• Clustering Distance measures, k-means clustering, nearest neighbor clustering</li> <li>• Kernel Density Estimation</li> <li>• Ensemble Learning</li> <li>• Reinforcement Learning</li> <li>• Computational Learning Theory</li> </ul>
<b>Literature</b>	<ol style="list-style-type: none"> <li>1. Artificial Intelligence: A Modern Approach (Third Edition), Stuart Russel, Peter Norvig, Prentice Hall, 2010, Chapters 13, 14, 18-21</li> <li>2. Machine Learning: A Probabilistic Perspective, Kevin Murphy, MIT Press 2012</li> </ol>

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

Module M0623: Intelligent Systems in Medicine				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
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
<b>Module Responsible</b>	Prof. Alexander Schlaefer			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• principles of math (algebra, analysis/calculus)</li> <li>• principles of stochastics</li> <li>• principles of programming, Java/C++ and R/Matlab</li> <li>• advanced programming skills</li> </ul>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>				
<i>Knowledge</i>	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.			
<i>Skills</i>	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.			
<b>Personal Competence</b>				
<i>Social Competence</i>	The students discuss the results of other groups, provide helpful feedback and can incorporate feedback into their work.			
<i>Autonomy</i>	The students can reflect their knowledge and document the results of their work. They can present the results in an appropriate manner.			
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	Yes	10 %	Presentation	
	Yes	10 %	Written elaboration	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 minutes			
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation II: Intelligence Engineering: 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 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: Technical Complementary Course: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Bio- and Medical Technology: Elective Compulsory			

Course L0331: Intelligent Systems in Medicine	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Alexander Schlaefer
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	- 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.
<b>Literature</b>	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	
<b>Typ</b>	Project Seminar
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Alexander Schlaefer
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

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



Module M1302: Applied Humanoid Robotics			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Applied Humanoid Robotics (L1794)	Project-/problem-based Learning	6	6
<b>Module Responsible</b>	Patrick Götttsch		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Object oriented programming; algorithms and data structures</li> <li>• Introduction to control systems</li> <li>• Control systems theory and design</li> <li>• Mechanics</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<ul style="list-style-type: none"> <li>• Students can explain humanoid robots.</li> <li>• Students can explain the basic concepts, relationships and methods of forward- and inverse kinematics</li> <li>• Students learn to apply basic control concepts for different tasks in humanoid robotics.</li> </ul>		
<i>Knowledge</i>			
<i>Skills</i>	<ul style="list-style-type: none"> <li>• Students can implement models for humanoid robotic systems in Matlab and C++, and use these models for robot motion or other tasks.</li> <li>• They are capable of using models in Matlab for simulation and testing these models if necessary with C++ code on the real robot system.</li> <li>• They are capable of selecting methods for solving abstract problems, for which no standard methods are available, and apply it successfully.</li> </ul>		
<b>Personal Competence</b>	<ul style="list-style-type: none"> <li>• Students can develop joint solutions in mixed teams and present these.</li> <li>• They can provide appropriate feedback to others, and constructively handle feedback on their own results</li> </ul>		
<i>Social Competence</i>			
<i>Autonomy</i>	<ul style="list-style-type: none"> <li>• Students are able to obtain required information from provided literature sources, and to put in into the context of the lecture.</li> <li>• They can independently define tasks and apply the appropriate means to solve them.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 96, Study Time in Lecture 84		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written elaboration		
<b>Examination duration and scale</b>	5-10 pages		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation II: Intelligence Engineering: Elective Compulsory Electrical Engineering: Specialisation Control and Power Systems Engineering: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Bio- and Medical Technology: Elective Compulsory Theoretical Mechanical Engineering: Technical Complementary Course: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory		
Course L1794: Applied Humanoid Robotics			
<b>Typ</b>	Project-/problem-based Learning		
<b>Hrs/wk</b>	6		
<b>CP</b>	6		
<b>Workload in Hours</b>	Independent Study Time 96, Study Time in Lecture 84		
<b>Lecturer</b>	Patrick Götttsch		
<b>Language</b>	DE/EN		
<b>Cycle</b>	WiSe/SoSe		
<b>Content</b>	<ul style="list-style-type: none"> <li>• Fundamentals of kinematics</li> <li>• Static and dynamic stability of humanoid robotic systems</li> <li>• Combination of different software environments (Matlab, C++, etc.)</li> <li>• Introduction to the necessary software frameworks</li> <li>• Team project</li> <li>• Presentation and Demonstration of intermediate and final results</li> </ul>		
<b>Literature</b>	<ul style="list-style-type: none"> <li>• B. Siciliano, O. Khatib. "Handbook of Robotics. Part A: Robotics Foundations", Springer (2008)</li> </ul>		

Module M1249: Medical Imaging			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Medical Imaging (L1694)	Lecture	2	3
Medical Imaging (L1695)	Recitation Section (small)	2	3
<b>Module Responsible</b>	Prof. Tobias Knopp		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Basic knowledge in linear algebra, numerics, and signal processing		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> 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.</p> <p><i>Skills</i> 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.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> 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.</p> <p><i>Autonomy</i> Students are able to independently investigate a complex problem and assess which competencies are required to solve it.</p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation II: Intelligence Engineering: Elective Compulsory Electrical Engineering: Specialisation Medical Technology: Elective Compulsory Interdisciplinary Mathematics: Specialisation Computational Methods in Biomedical Imaging: Compulsory Microelectronics and Microsystems: Specialisation Communication and Signal Processing: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Bio- and Medical Technology: Elective Compulsory Theoretical Mechanical Engineering: Technical Complementary Course: Elective Compulsory		

Course L1694: Medical Imaging	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Tobias Knopp
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Overview about different imaging methods</li> <li>• Signal processing</li> <li>• Inverse problems</li> <li>• Computed tomography</li> <li>• Magnetic resonance imaging</li> <li>• Compressed Sensing</li> <li>• Magnetic particle imaging</li> </ul>
<b>Literature</b>	<p><b>Bildgebende Verfahren in der Medizin</b>; O. Dössel; Springer, Berlin, 2000</p> <p><b>Bildgebende Systeme für die medizinische Diagnostik</b>; H. Morneburg (Hrsg.); Publicis MCD, München, 1995</p> <p><b>Introduction to the Mathematics of Medical Imaging</b>; C. L. Epstein; Siam, Philadelphia, 2008</p> <p><b>Medical Image Processing, Reconstruction and Restoration</b>; J. Jan; Taylor and Francis, Boca Raton, 2006</p> <p><b>Principles of Magnetic Resonance Imaging</b>; Z.-P. Liang and P. C. Lauterbur; IEEE Press, New York, 1999</p>

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

**Specialization III. Mathematics**

<b>Module M0667: Algorithmic Algebra</b>			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Algorithmic Algebra (L0422)	Lecture	3	5
Algorithmic Algebra (L0423)	Recitation Section (small)	1	1
<b>Module Responsible</b>	Dr. Prashant Batra		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Mathe I-III (Real analysis, computing in Vector spaces , principle of complete induction) Diskrete Mathematik I (groupus, rings, ideals, fields; euclidean algorithm)		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> 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.</p> <p><i>Skills</i> Students are able to access independently further logical connections between the concepts with which they have become familiar and are able to verify them.</p> <p>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.</p>		
<b>Personal Competence</b>	<p><i>Social Competence</i></p> <p><i>Autonomy</i></p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	30 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation III. Mathematics: Elective Compulsory		

Course L0422: Algorithmic Algebra	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	5
<b>Workload in Hours</b>	Independent Study Time 108, Study Time in Lecture 42
<b>Lecturer</b>	Dr. Prashant Batra
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<p>Extended euclidean algorithm, solution of the Bezout-equation</p> <p>Division with remainder (over rings)</p> <p>fast arithmetic algorithms (conversion, fast multiplications)</p> <p>discrete Fourier-transformation over rings</p> <p>Computation with modular remainders, solving of remainder systems (chinese remainder theorem), solvability of integer linear systems over the integers</p> <p>linearization of polynomial equations-- matrix approach</p> <p>Sylvester-matrix, elimination</p> <p>elimination in rings, elimination of many variables</p> <p>Buchberger algorithm, Gröbner basis</p> <p>Minkowskis Lattice Point theorem and integer-valued optimization</p> <p>LLL-algorithm for construction of 'short' lattice vectors in polynomial time</p>
<b>Literature</b>	<p>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).</p> <p>Yap, Chee Keng                      Fundamental problems of algorithmic algebra. (English) Zbl 0999.68261                      Oxford: Oxford University Press. xvi, 511 p. \$ 87.00 (2000).</p> <p>Free download for students from author's website: <a href="http://cs.nyu.edu/yap/book/berlin/">http://cs.nyu.edu/yap/book/berlin/</a></p> <p>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: <a href="http://dx.doi.org/10.1007/978-0-387-35651-8">http://dx.doi.org/10.1007/978-0-387-35651-8</a></p> <p style="text-align: right;">Concrete abstract algebra : from numbers to Gröbner bases /                      Niels <b>Lauritzen</b>                      Lauritzen, Niels                      Reprinted with corr.                      Cambridge [u.a.] : Cambridge Univ. Press, 2006                      XIV, 240 S. : graph. Darst.                      Includes bibliographical references and index                      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</p> <p>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: <a href="http://dx.doi.org/10.1007/3-540-29895-9">http://dx.doi.org/10.1007/3-540-29895-9</a></p> <p>Kaplan, Michael                      Computer algebra. (Computeralgebra.) (German) Zbl 1093.68148                      Berlin: Springer (ISBN 3-540-21379-1/pbk). xii, 391 p.                      springer eBook:  <a href="http://dx.doi.org/10.1007/b137968">http://dx.doi.org/10.1007/b137968</a></p>

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

Module M1428: Linear and Nonlinear Optimization			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Linear and Nonlinear Optimization (L2062)	Lecture	4	4
Linear and Nonlinear Optimization (L2063)	Recitation Section (large)	1	2
<b>Module Responsible</b>	Prof. Matthias Mnich		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Discrete Algebraic Structures</li> <li>Mathematics I</li> <li>Graph Theory and Optimization</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<ul style="list-style-type: none"> <li>Students can name the basic concepts in linear and non-linear optimization. They are able to explain them using appropriate examples.</li> <li>Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>They know proof strategies and can reproduce them.</li> </ul>		
<i>Knowledge</i>			
<i>Skills</i>	<ul style="list-style-type: none"> <li>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.</li> <li>Students are able to discover and verify further logical connections between the concepts studied in the course.</li> <li>For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.</li> </ul>		
<b>Personal Competence</b>	<ul style="list-style-type: none"> <li>Students are able to work together in teams. They are capable to use mathematics as a common language.</li> <li>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.</li> </ul>		
<i>Social Competence</i>			
<i>Autonomy</i>	<ul style="list-style-type: none"> <li>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.</li> <li>Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	30 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation III. Mathematics: Elective Compulsory Computational Science and Engineering: Specialisation III. Mathematics: Elective Compulsory		
Course L2062: Linear and Nonlinear Optimization			
<b>Typ</b>	Lecture		
<b>Hrs/wk</b>	4		
<b>CP</b>	4		
<b>Workload in Hours</b>	Independent Study Time 64, Study Time in Lecture 56		
<b>Lecturer</b>	Prof. Matthias Mnich		
<b>Language</b>	DE/EN		
<b>Cycle</b>	WiSe		
<b>Content</b>	<ul style="list-style-type: none"> <li>Modelling linear programming problems</li> <li>Graphical method</li> <li>Algebraic background</li> <li>Convexity</li> <li>Polyhedral theory</li> <li>Simplex method</li> <li>Degeneracy and convergence</li> <li>duality</li> <li>interior-point methods</li> <li>quadratic optimization</li> <li>integer linear programming</li> </ul>		
<b>Literature</b>	<ul style="list-style-type: none"> <li>A. Schrijver: Combinatorial Optimization: Polyhedra and Efficiency. Springer, 2003</li> <li>B. Korte and T. Vygen: Combinatorial Optimization: Theory and Algorithms. Springer, 2018</li> <li>T. Cormen, Ch. Leiserson, R. Rivest, C. Stein: Introduction to Algorithms. MIT Press, 2013</li> </ul>		

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



Module M0716: Hierarchical Algorithms			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Hierarchical Algorithms (L0585)	Lecture	2	3
Hierarchical Algorithms (L0586)	Recitation Section (small)	2	3
<b>Module Responsible</b>	Prof. Sabine Le Borne		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Mathematics I, II, III for Engineering students (german or english) or Analysis &amp; Linear Algebra I + II as well as Analysis III for Technomathematicians</li> <li>Programming experience in C</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> Students are able to</p> <ul style="list-style-type: none"> <li>name representatives of hierarchical algorithms and list their characteristics,</li> <li>explain construction techniques for hierarchical algorithms,</li> <li>discuss aspects regarding the efficient implementation of hierarchical algorithms.</li> </ul> <p><i>Skills</i> Students are able to</p> <ul style="list-style-type: none"> <li>implement the hierarchical algorithms discussed in the lecture,</li> <li>analyse the storage and computational complexities of the algorithms,</li> <li>adapt algorithms to problem settings of various applications and thus develop problem adapted variants.</li> </ul> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> Students are able to</p> <ul style="list-style-type: none"> <li>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.</li> </ul> <p><i>Autonomy</i> Students are capable</p> <ul style="list-style-type: none"> <li>to assess whether the supporting theoretical and practical exercises are better solved individually or in a team,</li> <li>to work on complex problems over an extended period of time,</li> <li>to assess their individual progress and, if necessary, to ask questions and seek help.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	20 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation III. Mathematics: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Technical Complementary Course: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Simulation Technology: Elective Compulsory		
Course L0585: Hierarchical Algorithms			
<b>Typ</b>	Lecture		
<b>Hrs/wk</b>	2		
<b>CP</b>	3		
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28		
<b>Lecturer</b>	Prof. Sabine Le Borne		
<b>Language</b>	DE/EN		
<b>Cycle</b>	WiSe		
<b>Content</b>	<ul style="list-style-type: none"> <li>Low rank matrices</li> <li>Separable expansions</li> <li>Hierarchical matrix partitions</li> <li>Hierarchical matrices</li> <li>Formatted matrix operations</li> <li>Applications</li> <li>Additional topics (e.g. H2 matrices, matrix functions, tensor products)</li> </ul>		
<b>Literature</b>	W. Hackbusch: Hierarchische Matrizen: Algorithmen und Analysis		

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

Module M1405: Randomised Algorithms and Random Graphs			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Randomised Algorithms and Random Graphs (L2010)	Lecture	2	3
Randomised Algorithms and Random Graphs (L2011)	Recitation Section (large)	2	3
<b>Module Responsible</b>	Prof. Anusch Taraz		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b> <i>Knowledge</i>	<ul style="list-style-type: none"> <li>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.</li> <li>Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>They know proof strategies and can apply them.</li> </ul>		
<i>Skills</i>	<ul style="list-style-type: none"> <li>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.</li> <li>Students are able to explore and verify further logical connections between the concepts studied in the course.</li> <li>For a given problem, the students can develop and execute a suitable technique, and are able to critically evaluate the results.</li> </ul>		
<b>Personal Competence</b> <i>Social Competence</i>	<ul style="list-style-type: none"> <li>Students are able to work together in teams. They are capable to establish a common language.</li> <li>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.</li> </ul>		
<i>Autonomy</i>	<ul style="list-style-type: none"> <li>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.</li> <li>Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	30 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation III. Mathematics: Elective Compulsory Computational Science and Engineering: Specialisation III. Mathematics: Elective Compulsory		

Course L2010: Randomised Algorithms and Random Graphs	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Anusch Taraz, Prof. Volker Turau
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<p>Randomized Algorithms:</p> <ul style="list-style-type: none"> <li>• introduction and recalling basic tools from probability</li> <li>• randomized search</li> <li>• random walks</li> <li>• text search with fingerprinting</li> <li>• parallel and distributed algorithms</li> <li>• online algorithms</li> </ul> <p>Random Graphs:</p> <ul style="list-style-type: none"> <li>• typical properties</li> <li>• first and second moment method</li> <li>• tail bounds</li> <li>• thresholds and phase transitions</li> <li>• probabilistic method</li> <li>• models for complex networks</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Motwani, Raghavan: Randomized Algorithms</li> <li>• Worsch: Randomisierte Algorithmen</li> <li>• Dietzfelbinger: Randomisierte Algorithmen</li> <li>• Bollobas: Random Graphs</li> <li>• Alon, Spencer: The Probabilistic Method</li> <li>• Frieze, Karonski: Random Graphs</li> <li>• van der Hofstad: Random Graphs and Complex Networks</li> </ul>

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

Module M0714: Numerical Treatment of Ordinary Differential Equations			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Numerical Treatment of Ordinary Differential Equations (L0576)		Lecture	2
Numerical Treatment of Ordinary Differential Equations (L0582)		Recitation Section (small)	2
<b>CP</b>			
			3
<b>Module Responsible</b>	Prof. Daniel Ruprecht		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Mathematik I, II, III für Ingenieurstudierende (deutsch oder englisch) oder Analysis &amp; Lineare Algebra I + II sowie Analysis III für Technomathematiker</li> <li>Basic MATLAB knowledge</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	Students are able to <ul style="list-style-type: none"> <li>list numerical methods for the solution of ordinary differential equations and explain their core ideas,</li> <li>repeat convergence statements for the treated numerical methods (including the prerequisites tied to the underlying problem),</li> <li>explain aspects regarding the practical execution of a method.</li> <li>select the appropriate numerical method for concrete problems, implement the numerical algorithms efficiently and interpret the numerical results</li> </ul>		
<i>Skills</i>	Students are able to <ul style="list-style-type: none"> <li>implement (MATLAB), apply and compare numerical methods for the solution of ordinary differential equations,</li> <li>to justify the convergence behaviour of numerical methods with respect to the posed problem and selected algorithm,</li> <li>for a given problem, develop a suitable solution approach, if necessary by the composition of several algorithms, to execute this approach and to critically evaluate the results.</li> </ul>		
<b>Personal Competence</b>			
<i>Social Competence</i>	Students are able to <ul style="list-style-type: none"> <li>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.</li> </ul>		
<i>Autonomy</i>	Students are capable <ul style="list-style-type: none"> <li>to assess whether the supporting theoretical and practical exercises are better solved individually or in a team,</li> <li>to assess their individual progress and, if necessary, to ask questions and seek help.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 min		
<b>Assignment for the Following Curricula</b>	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	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Daniel Ruprecht
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<p>Numerical methods for Initial Value Problems</p> <ul style="list-style-type: none"> <li>• single step methods</li> <li>• multistep methods</li> <li>• stiff problems</li> <li>• differential algebraic equations (DAE) of index 1</li> </ul> <p>Numerical methods for Boundary Value Problems</p> <ul style="list-style-type: none"> <li>• multiple shooting method</li> <li>• difference methods</li> <li>• variational methods</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• E. Hairer, S. Noersett, G. Wanner: Solving Ordinary Differential Equations I: Nonstiff Problems</li> <li>• E. Hairer, G. Wanner: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems</li> </ul>

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

Module M1668: Probability Theory			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Probability Theory (L2643)		Lecture	3
Probability Theory (L2644)		Recitation Section (small)	1
<b>CP</b>			
			4
			2
<b>Module Responsible</b>	Prof. Matthias Schulte		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Familiarity with the basic concepts of probability		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<ul style="list-style-type: none"> <li>• Students can name the basic concepts in probability theory. They are able to explain them using appropriate examples.</li> <li>• Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>• They know proof strategies and can reproduce them.</li> </ul>		
<i>Knowledge</i>			
<i>Skills</i>			
<b>Personal Competence</b>			
<i>Social Competence</i>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• Students are able to explore and verify further logical connections between the concepts studied in the course.</li> <li>• For a given problem, the students can develop and execute a suitable technique, and are able to critically evaluate the results.</li> </ul>		
<i>Autonomy</i>	<ul style="list-style-type: none"> <li>• Students are able to work together (e.g. on their regular home work) and to present their results appropriately (e.g. during exercise class).</li> <li>• 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.</li> </ul>		
	<ul style="list-style-type: none"> <li>• 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.</li> <li>• Students can put their knowledge in relation to the contents of other lectures.</li> <li>• Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	30 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation III. Mathematics: Elective Compulsory Interdisciplinary Mathematics: Specialisation II. Numerical - Modelling Training: Compulsory		
Course L2643: Probability Theory			
<b>Typ</b>	Lecture		
<b>Hrs/wk</b>	3		
<b>CP</b>	4		
<b>Workload in Hours</b>	Independent Study Time 78, Study Time in Lecture 42		
<b>Lecturer</b>	Prof. Matthias Schulte		
<b>Language</b>	EN		
<b>Cycle</b>	SoSe		
<b>Content</b>	<ul style="list-style-type: none"> <li>• Measure and probability spaces</li> <li>• Integration and expectation</li> <li>• Types of stochastic convergence</li> <li>• Law of large numbers</li> <li>• Central limit theorem</li> <li>• Radon-Nikodym theorem</li> <li>• Conditional expectation</li> <li>• Martingales</li> <li>• Markov chains</li> <li>• Poisson processes</li> </ul>		
<b>Literature</b>	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.		

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



Module M0711: Numerical Mathematics II			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Numerical Mathematics II (L0568)	Lecture	2	3
Numerical Mathematics II (L0569)	Recitation Section (small)	2	3
<b>Module Responsible</b>	Prof. Sabine Le Borne		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Numerical Mathematics I</li> <li>Python knowledge</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	<p>Students are able to</p> <ul style="list-style-type: none"> <li>name advanced numerical methods for interpolation, approximation, integration, eigenvalue problems, eigenvalue problems, nonlinear root finding problems and explain their core ideas,</li> <li>repeat convergence statements for the numerical methods, sketch convergence proofs,</li> <li>explain practical aspects of numerical methods concerning runtime and storage needs</li> <li>explain aspects regarding the practical implementation of numerical methods with respect to computational and storage complexity.</li> </ul>		
<i>Skills</i>	<p>Students are able to</p> <ul style="list-style-type: none"> <li>implement, apply and compare advanced numerical methods in Python,</li> <li>justify the convergence behaviour of numerical methods with respect to the problem and solution algorithm and to transfer it to related problems,</li> <li>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</li> </ul>		
<b>Personal Competence</b>			
<i>Social Competence</i>	<p>Students are able to</p> <ul style="list-style-type: none"> <li>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.</li> </ul>		
<i>Autonomy</i>	<p>Students are capable</p> <ul style="list-style-type: none"> <li>to assess whether the supporting theoretical and practical exercises are better solved individually or in a team,</li> <li>to assess their individual progress and, if necessary, to ask questions and seek help.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	25 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation III. Mathematics: Elective Compulsory Computational Science and Engineering: Specialisation III. Mathematics: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Technical Complementary Course: Elective Compulsory Theoretical Mechanical Engineering: Core qualification: Elective Compulsory		

Course L0568: Numerical Mathematics II	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sabine Le Borne, Dr. Jens-Peter Zemke
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ol style="list-style-type: none"> <li>1. Error and stability: Notions and estimates</li> <li>2. Rational interpolation and approximation</li> <li>3. Multidimensional interpolation (RBF) and approximation (neural nets)</li> <li>4. Quadrature: Gaussian quadrature, orthogonal polynomials</li> <li>5. Linear systems: Perturbation theory of decompositions, structured matrices</li> <li>6. Eigenvalue problems: LR-, QD-, QR-Algorithmus</li> <li>7. Nonlinear systems of equations: Newton and Quasi-Newton methods, line search (optional)</li> <li>8. Krylov space methods: Arnoldi-, Lanczos methods (optional)</li> </ol>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Skript</li> <li>• Stoer/Bulirsch: Numerische Mathematik 1, Springer</li> <li>• Dahmen, Reusken: Numerik für Ingenieure und Naturwissenschaftler, Springer</li> </ul>

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

Module M0881: Mathematical Image Processing			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Mathematical Image Processing (L0991)		Lecture	3
Mathematical Image Processing (L0992)		Recitation Section (small)	1
<b>CP</b>			2
<b>Module Responsible</b>	Prof. Marko Lindner		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Analysis: partial derivatives, gradient, directional derivative</li> <li>• Linear Algebra: eigenvalues, least squares solution of a linear system</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <p>Students are able to</p> <ul style="list-style-type: none"> <li>• characterize and compare diffusion equations</li> <li>• explain elementary methods of image processing</li> <li>• explain methods of image segmentation and registration</li> <li>• sketch and interrelate basic concepts of functional analysis</li> </ul> <p><i>Skills</i></p> <p>Students are able to</p> <ul style="list-style-type: none"> <li>• implement and apply elementary methods of image processing</li> <li>• explain and apply modern methods of image processing</li> </ul>		
<b>Personal Competence</b>	<p><i>Social Competence</i></p> <p>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.</p> <p><i>Autonomy</i></p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	20 min		
<b>Assignment for the Following Curricula</b>	Bioprocess Engineering: Specialisation A - General Bioprocess Engineering: Elective Compulsory Computer Science: Specialisation III. Mathematics: Elective Compulsory Computational Science and Engineering: Specialisation III. Mathematics: Elective Compulsory Interdisciplinary Mathematics: Specialisation Computational Methods in Biomedical Imaging: Compulsory Mechatronics: Technical Complementary Course: Elective Compulsory Mechatronics: Specialisation System Design: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Technical Complementary Course: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory Process Engineering: Specialisation Process Engineering: Elective Compulsory		

Course L0991: Mathematical Image Processing	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 78, Study Time in Lecture 42
<b>Lecturer</b>	Prof. Marko Lindner
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• basic methods of image processing</li> <li>• smoothing filters</li> <li>• the diffusion / heat equation</li> <li>• variational formulations in image processing</li> <li>• edge detection</li> <li>• de-convolution</li> <li>• inpainting</li> <li>• image segmentation</li> <li>• image registration</li> </ul>
<b>Literature</b>	Bredies/Lorenz: Mathematische Bildverarbeitung

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

Module M1552: Mathematics of Neural Networks			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Mathematics of Neural Networks (L2322)		Lecture	2
Mathematics of Neural Networks (L2323)		Recitation Section (small)	2
<b>CP</b>			
			3
<b>Module Responsible</b>	Dr. Jens-Peter Zemke		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ol style="list-style-type: none"> <li>1. Mathematics I-III</li> <li>2. Numerical Mathematics 1/ Numerics</li> <li>3. Programming skills, preferably in Python</li> </ol>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> 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.</p> <p><i>Skills</i> Students are able to implement, understand, and, tailored to the field of application, apply neural networks.</p>		
<b>Personal Competence</b>	<p><i>Social Competence</i> Students can</p> <ul style="list-style-type: none"> <li>• develop and document joint solutions in small teams;</li> <li>• form groups to further develop the ideas and transfer them to other areas of applicability;</li> <li>• form a team to develop, build, and advance a software library.</li> </ul> <p><i>Autonomy</i> Students are able to</p> <ul style="list-style-type: none"> <li>• correctly assess the time and effort of self-defined work;</li> <li>• assess whether the supporting theoretical and practical exercises are better solved individually or in a team;</li> <li>• define test problems for testing and expanding the methods;</li> <li>• assess their individual progress and, if necessary, to ask questions and seek help.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	25 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation III. Mathematics: Elective Compulsory Computational Science and Engineering: Specialisation III. Mathematics: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Mechatronics: Technical Complementary Course: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory		

Course L2322: Mathematics of Neural Networks	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Dr. Jens-Peter Zemke
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ol style="list-style-type: none"> <li>1. Basics: analogy; layout of neural nets, universal approximation, NP-completeness</li> <li>2. Feedforward nets: backpropagation, variants of Stochastic Gradients</li> <li>3. Deep Learning: problems and solution strategies</li> <li>4. Deep Belief Networks: energy based models, Contrastive Divergence</li> <li>5. CNN: idea, layout, FFT and Winograds algorithms, implementation details</li> <li>6. RNN: idea, dynamical systems, training, LSTM</li> <li>7. ResNN: idea, relation to neural ODEs</li> <li>8. Standard libraries: Tensorflow, Keras, PyTorch</li> <li>9. Recent trends</li> </ol>
<b>Literature</b>	<ol style="list-style-type: none"> <li>1. Skript</li> <li>2. Online-Werke: <ul style="list-style-type: none"> <li>◦ <a href="http://neuralnetworksanddeeplearning.com/">http://neuralnetworksanddeeplearning.com/</a></li> <li>◦ <a href="https://www.deeplearningbook.org/">https://www.deeplearningbook.org/</a></li> </ul> </li> </ol>

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

Module M1020: Numerics of Partial Differential Equations			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Numerics of Partial Differential Equations (L1247)	Lecture	2	3
Numerics of Partial Differential Equations (L1248)	Recitation Section (small)	2	3
<b>Module Responsible</b>	Prof. Daniel Ruprecht		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Mathematik I - IV (for Engineering Students) <b>or</b> Analysis &amp; Linear Algebra I + II for Technomathematicians</li> <li>• Numerical mathematics 1</li> <li>• Numerical treatment of ordinary differential equations</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <ul style="list-style-type: none"> <li>• Students can classify partial differential equations according to the three basic types.</li> <li>• For each type, students know suitable numerical approaches.</li> <li>• Students know the theoretical convergence results for these approaches.</li> </ul> <p><i>Skills</i></p> <p>Students are capable to formulate solution strategies for given problems involving partial differential equations, to comment on theoretical properties concerning convergence and to implement and test these methods in practice.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p>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.</p> <p><i>Autonomy</i></p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	25 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation III. Mathematics: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Technical Complementary Course: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Simulation Technology: Elective Compulsory		

Course L1247: Numerics of Partial Differential Equations	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Daniel Ruprecht
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	Elementary Theory and Numerics of PDEs <ul style="list-style-type: none"> <li>• types of PDEs</li> <li>• well posed problems</li> <li>• finite differences</li> <li>• finite elements</li> <li>• finite volumes</li> <li>• applications</li> </ul>
<b>Literature</b>	Dietrich Braess: Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie, Berlin u.a., Springer 2007  Susanne Brenner, Ridgway Scott: The Mathematical Theory of Finite Element Methods, Springer, 2008  Peter Deuffhard, Martin Weiser: Numerische Mathematik 3

<b>Course L1248: Numerics of Partial Differential Equations</b>	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Daniel Ruprecht
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course



---



---

**Specialization IV. Subject Specific Focus**


---



---

**Module M1565: Technical Complementary Course I for CSMS**
**Courses**

<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
<b>Module Responsible</b>	Prof. Karl-Heinz Zimmermann		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b> <i>Knowledge</i> <i>Skills</i> <b>Personal Competence</b> <i>Social Competence</i> <i>Autonomy</i>			
<b>Workload in Hours</b>	Depends on choice of courses		
<b>Credit points</b>	6		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation IV. Subject Specific Focus: Elective Compulsory		

Module M1566: Technical Complementary Course II for CSMS				
Courses				
Title		Typ	Hrs/wk	CP
<b>Module Responsible</b>	Prof. Karl-Heinz Zimmermann			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>				
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b> <i>Knowledge</i> <i>Skills</i>				
<b>Personal Competence</b> <i>Social Competence</i> <i>Autonomy</i>				
<b>Workload in Hours</b>	Depends on choice of courses			
<b>Credit points</b>	6			
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation IV. Subject Specific Focus: Elective Compulsory			

## Thesis

Module M-002: Master Thesis			
Courses			
Title	Typ	Hrs/wk	CP
<b>Module Responsible</b>	Professoren der TUHH		
<b>Admission Requirements</b>	<ul style="list-style-type: none"> <li>According to General Regulations §21 (1):</li> </ul> <p>At least 60 credit points have to be achieved in study programme. The examinations board decides on exceptions.</p>		
<b>Recommended Previous Knowledge</b>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <ul style="list-style-type: none"> <li>The students can use specialized knowledge (facts, theories, and methods) of their subject competently on specialized issues.</li> <li>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.</li> <li>The students can place a research task in their subject area in its context and describe and critically assess the state of research.</li> </ul> <p><i>Skills</i></p> <p>The students are able:</p> <ul style="list-style-type: none"> <li>To select, apply and, if necessary, develop further methods that are suitable for solving the specialized problem in question.</li> <li>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.</li> <li>To develop new scientific findings in their subject area and subject them to a critical assessment.</li> </ul> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p>Students can</p> <ul style="list-style-type: none"> <li>Both in writing and orally outline a scientific issue for an expert audience accurately, understandably and in a structured way.</li> <li>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.</li> </ul> <p><i>Autonomy</i></p> <p>Students are able:</p> <ul style="list-style-type: none"> <li>To structure a project of their own in work packages and to work them off accordingly.</li> <li>To work their way in depth into a largely unknown subject and to access the information required for them to do so.</li> <li>To apply the techniques of scientific work comprehensively in research of their own.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 900, Study Time in Lecture 0		
<b>Credit points</b>	30		
<b>Course achievement</b>	None		
<b>Examination</b>	Thesis		
<b>Examination duration and scale</b>	According to General Regulations		
<b>Assignment for the Following Curricula</b>	Civil Engineering: Thesis: Compulsory Bioprocess Engineering: Thesis: Compulsory Chemical and Bioprocess Engineering: Thesis: Compulsory Computer Science: Thesis: Compulsory Electrical Engineering: Thesis: Compulsory Energy and Environmental Engineering: Thesis: Compulsory Energy Systems: Thesis: Compulsory Environmental Engineering: Thesis: Compulsory Aircraft Systems Engineering: Thesis: Compulsory Global Innovation Management: Thesis: Compulsory Computational Science and Engineering: Thesis: Compulsory Information and Communication Systems: Thesis: Compulsory Interdisciplinary Mathematics: 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 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