



## **Module Manual**

Bachelor of Science (B.Sc.)

# **Computational Science and Engineering**

Cohort: Winter Term 2021

Updated: 20th April 2023



# Table of Contents

Table of Contents	2
Program description	3
Core Qualification	5
Module M0561: Discrete Algebraic Structures	5
Module M1436: Procedural Programming for Computer Engineers	6
Module M0850: Mathematics I	7
Module M0577: Non-technical Courses for Bachelors	10
Module M0743: Electrical Engineering I: Direct Current Networks and Electromagnetic Fields	12
Module M0547: Electrical Engineering II: Alternating Current Networks and Basic Devices	13
Module M0624: Automata Theory and Formal Languages	16
Module M0829: Foundations of Management	18
Module M0851: Mathematics II	21
Module M1432: Programming Paradigms	24
Module M0834: Computernetworks and Internet Security	26
Module M0662: Numerical Mathematics I	28
Module M0730: Computer Engineering	30
Module M0853: Mathematics III	32
Module M1423: Algorithms and Data Structures	35
Module M1578: Seminars Computer Science	37
Module M0672: Signals and Systems	38
Module M0803: Embedded Systems	41
Module M0727: Stochastics	43
Module M0833: Introduction to Control Systems	45
Module M0675: Introduction to Communications and Random Processes	47
Module M1431: Practical Course IIW	51
Specialization I. Computer Science	52
Module M0731: Functional Programming	52
Module M0625: Databases	55
Module M0791: Computer Architecture	57
Module M1883: Introduction to Quantum Computing	59
Module M0562: Computability and Complexity Theory	61
Module M0754: Compiler Construction	63
Module M0732: Software Engineering	64
Module M1300: Software Development	66
Module M1595: Machine Learning I	68
Module M1908: Fundamentals of Operating Systems	70
Module M1872: Operating System Construction for Single-Core Systems	72
Specialization II. Mathematics & Engineering Science	74
Module M1235: Electrical Power Systems I: Introduction to Electrical Power Systems	74
Module M0760: Electronic Devices	77
Module M1896: Machine Dynamics	79
Module M0708: Electrical Engineering III: Circuit Theory and Transients	82
Module M0941: Combinatorial Structures and Algorithms	84
Module M1802: Engineering Mechanics I (Stereostatics)	86
Module M0783: Measurements: Methods and Data Processing	88
Module M1712: Green Technologies II	90
Module M0634: Introduction into Medical Technology and Systems	92
Module M0715: Solvers for Sparse Linear Systems	94
Module M0777: Semiconductor Circuit Design	96
Module M1269: Lab Cyber-Physical Systems	98
Module M0854: Mathematics IV	99
Module M0610: Electrical Machines and Actuators	102
Module M0567: Theoretical Electrical Engineering I: Time-Independent Fields	104
Specialization III. Subject Specific Focus	106
Module M1433: Technical Complementary Course for Computational Science and Engineering Bachelor	106
Thesis	107
Module M-001: Bachelor Thesis	107

---

---

## Program description

---

---

### Content

Engineering disciplines utilize the results of computer science and mathematics research to an ever greater extent, both in the development of products and in the products themselves. This trend will certainly continue. New results in computer science and mathematics thus become an important innovation factor in engineering and are therefore central areas of competence for an engineer and a technical university. This has a direct impact on the objectives of the computer science and engineering course.

Engineering education benefits significantly from computer science, and computer science benefits significantly from the modeling techniques used in engineering. To be prepared for the requirements of the future, the aim of the course is to offer combined training in computer science, mathematics and engineering. This is a particularly sustainable training principle, both for industry and for research. Computer engineering opens the line between hardware and software in the light of engineering applications. Decisions as to which parts of a system should be implemented more cheaply in hardware or better with the help of flexible software can only be made and carried out on the basis of solid knowledge of both disciplines, both IT and engineering. The aim of the course is to introduce the problem and to deal with both essential aspects.

The objectives of the basic qualification are to impart knowledge, skills and competences in the fields of computer science, mathematics and engineering to the students so that new areas of knowledge and thus also new products can be developed. Choices that support student in self-determined studies/instruction are offered in specialisation areas.

### Career prospects

Successful completion of the bachelor's degree in computer science engineering at TUHH enables graduates to start a career in science, computer science engineering or a related subject, as well as an early career start in areas from trade, industry and administration (professional qualification). The graduates will then primarily work as engineers and system developers for software and hardware.

Because of their broad training, graduates are particularly requested in the job market, since the bridge between IT specialists and engineers is essential in system development. Depending on the chosen specialization, the course trains computer scientists with an engineering background or engineers with a computer science background, who find very good employment opportunities on the German and international job market largely regardless of economic trends.

### Learning target

The learning objectives leading towards the described qualification are divided below into the categories knowledge, skills, social skills and independence.

#### Knowledge

The learned knowledge comprises facts, principles and theories in the subjects of computer science, engineering and mathematics.

1. Students can reproduce, define and explain known standard languages for representation used in computer science and mathematics (logic, automata theory, formal languages, graph theory, linear algebra, analysis, discrete algebraic structures, stochastics, systems theory, etc.) necessary for the formal modeling of application problems (syntax, semantics, decision problems).
2. Students can reproduce elementary data and index structures (vectors, matrices, relations, trees, files, pages) for sequential algorithms (also in hardware-related form) and show their advantages and disadvantages for special tasks. Students can specify algorithms to solve decision problems for formal modeling techniques. They can reproduce the basic structure of simple computing systems at different levels of abstraction in an architecture, so that you can explain how algorithms are executed on concrete systems.
3. The students are familiar with a whole range of classic applications of computer engineering and mathematical modeling techniques and can explain them.
4. Students know how problems can be broken down into smaller sub-problems (reductionist approach) and how partial results can be combined to form an overall result. Students can also describe problems that arise from error propagation and error accumulation and provide examples. Students can reproduce and justify that security, reliability, and maintenance of partial services in the event of an error (graceful degradation) can only result from concrete design decisions in an initial draft and cannot be integrated into an existing draft afterwards with reasonable effort.
5. Graduates are able to explain the importance of entrepreneurial planning and goals, to analyze the organizational and personnel structures as well as the production and procurement systems of companies, to classify pricing policy and other important instruments for system development (e.g. marketing).

#### Technical Skills

The course of Computer Science and Engineering teaches the ability to apply learned knowledge in order to complete tasks and thus solve problems in many facets.

1. Students can design and develop formal representation languages (syntax, semantics, decision problems), and they can assess and determine the expressiveness of the formalisms necessary for simple applications. Students can map decision problems of different formalisms onto one another and thus compare the expressiveness of formalisms.
2. Students can examine algorithms for decision problems for completeness and correctness or convergence behavior and approximation quality, and they can demonstrate whether an algorithm is optimal or for which types of inputs the worst case occurs with regard to the runtime behavior of an algorithm.
3. Students can implement algorithms in programming or hardware description languages, test them and integrate them into application systems using operating systems to manage resources and use databases to manage large amounts of data. Students can demonstrate that desired states of a system are reached (controllability, accessibility) and that undesired states are never reached (safety and liveness properties). Students can implement computer structures in hardware-related units.
4. Students can use formal modeling techniques for engineering applications to create, review, or evaluate simple, prototypical systems to solve problems from an application context (in terms of a simulation, as a data management system, as an application, etc.). Students can explain how models, programs and systems are automatically translated into corresponding units at a lower level of abstraction.
5. Students can design interfaces that allow systems to be built from modules or layers, the internals of which can be adapted without changing the interfaces. Students are able to describe design criteria, how systems can be reused and can also be used in other systems.

#### Social skills

The ability and the will to work with others in a goal-oriented manner, to grasp their interests and social situations, to communicate and to help shape the working and living environment is broken down as follows for the degree course in Computer Science and Engineering:

1. Students understand that methods of computer science and mathematics are developed across all applications and that a major achievement of the computer science engineer is on the one hand in the professional application of the methods and on the other hand in demonstrating others (clients, project partners, colleagues, ...) that a method is (in a specific sense) optimal.
2. Students can form teams to work in groups, define and distribute subtasks, make appointments, integrate partial solutions. They are able to communicate, interact socially and behave appropriately in the event of conflicts.
3. Students explain the problems described in a scientific paper and the solutions developed in the paper in a field of computer science or

# Module Manual B.Sc. "Computational Science and Engineering"

mathematics, evaluate the proposed solutions in a lecture and respond to scientific questions, additions and comments.

4. Students describe scientific questions in a field of computer science, engineering or mathematics and explain in a presentation an approach they have developed to solve it and respond appropriately to inquiries, additions and comments.

## Competence to work independently

The ability and willingness to act independently and responsibly, to reflect on one's own actions and those of others, and also to further develop one's own ability to act, is broken down as follows into finer aspects.

1. The students independently evaluate the advantages and disadvantages of representation formalisms for specific tasks, compare different algorithms and data structures as well as programming languages and programming tools, and they independently select the best solution.
2. The graduates independently develop a small, very clearly defined scientific sub-area, can present it in a presentation and actively follow the presentations of other students, so that an interactive discourse on a scientific topic arises.
3. Students integrate themselves into a project context and assume responsibility for tasks in a software or hardware development project.

## Program structure

The curriculum of the Bachelor's degree in Computer Science and Engineering is structured as follows. In addition to the compulsory courses from core qualification, a minimum number of credit points must be taken from each of the areas of computer science, mathematics and engineering:

1. Core qualification: 138 credit points
2. Computer science: 12 credit
3. Mathematics & Engineering: 6 credit points

To deepen their studies, students can choose lectures from the entire catalog of technical events at the TUHH. A total of 12 credit points must be achieved. The bachelor thesis is also rated with 12 credit points. This results in a total effort of 180 credit points.

The following four course plans describe special features of the IiW Bachelor's degree

### E. Embedded systems

1. Core subjects in computer science
  - Computer architecture
  - Operating systems
2. Core subjects: mathematics and engineering
  - Electronic components
3. Additional technical courses
  - Semiconductor circuit technology
  - Compiler construction

### I. Smart grids

1. Core subjects in computer science
  - Operating systems
  - Software development
2. Core subjects: mathematics and engineering
  - Electrical energy systems I
3. Additional technical courses
  - Theoretical electrical engineering I
  - Electrical engineering III: network theory and transients

### M. Medical systems

1. Core subjects in computer science
  - Introduction to information security
  - Software engineering
2. Core subjects: mathematics and engineering
  - Introduction to medical technology systems
3. Additional technical courses
  - Cyber-physical systems laboratory
  - Computer architecture

### C. Computational Foundations

1. Core subjects in computer science
  - Functional programming
  - Predictability and complexity
2. Core subjects: mathematics and engineering
  - Combinatorial structures and algorithms
3. Additional technical courses
  - Solvers for sparse linear equation systems
  - Mathematics IV

## Core Qualification

### Module M0561: Discrete Algebraic Structures

#### Courses

Title	Typ	Hrs/wk	CP
Discrete Algebraic Structures (L0164)	Lecture	2	3
Discrete Algebraic Structures (L0165)	Recitation Section (small)	2	3
<b>Module Responsible</b>	Prof. Karl-Heinz Zimmermann		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Mathematics from High School.		
<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 know the important basics of discrete algebraic structures including elementary combinatorial structures, monoids, groups, rings, fields, finite fields, and vector spaces. They also know specific structures like sub-, sum-, and quotient structures and homomorphisms.</p> <p><i>Skills</i> Students are able to formalize and analyze basic discrete algebraic structures.</p> <p><i>Personal Competence</i></p> <p><i>Social Competence</i> Students are able to solve specific 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 standard books and to associate the acquired knowledge to 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>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Computational Science and Engineering: Core Qualification: Compulsory Orientation Studies: Core Qualification: Elective Compulsory		

#### Course L0164: Discrete Algebraic Structures

<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. Karl-Heinz Zimmermann
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	
<b>Literature</b>	

#### Course L0165: Discrete Algebraic Structures

<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. Karl-Heinz Zimmermann
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

## Module M1436: Procedural Programming for Computer Engineers

### Courses

Title	Typ	Hrs/wk	CP
Procedural Programming for Computer Engineers (L2163)	Lecture	1	2
Procedural Programming for Computer Engineers (L2164)	Recitation Section (large)	1	1
Procedural Programming for Computer Engineers (L2165)	Practical Course	2	3
<b>Module Responsible</b>	NN		
<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: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Computational Science and Engineering: Core Qualification: Compulsory Technomathematics: Core Qualification: Compulsory		

### Course L2163: Procedural Programming for Computer Engineers

<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. Siegfried Rump
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	
<b>Literature</b>	

### Course L2164: Procedural Programming for Computer Engineers

<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>	Dozenten des SD E
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	
<b>Literature</b>	

### Course L2165: Procedural Programming for Computer Engineers

<b>Typ</b>	Practical Course
<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>	Dozenten des SD E
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	
<b>Literature</b>	

Module M0850: Mathematics I				
Courses				
Title		Type	Hrs/wk	CP
Analysis I (L1010)		Lecture	2	2
Analysis I (L1012)		Recitation Section (small)	1	1
Analysis I (L1013)		Recitation Section (large)	1	1
Linear Algebra I (L0912)		Lecture	2	2
Linear Algebra I (L0913)		Recitation Section (small)	1	1
Linear Algebra I (L0914)		Recitation Section (large)	1	1
Module Responsible	Prof. Anusch Taraz			
Admission Requirements	None			
Recommended Previous Knowledge	School mathematics			
Educational Objectives	After taking part successfully, students have reached the following learning results			
Professional Competence	<ul style="list-style-type: none"><li>Students can name the basic concepts in analysis and linear algebra. 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> <ul style="list-style-type: none"><li>Students can model problems in analysis and linear algebra 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> <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> <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>			
Knowledge				
Skills				
Personal Competence				
Social Competence				
Autonomy				
Workload in Hours	Independent Study Time 128, Study Time in Lecture 112			
Credit points	8			
Course achievement	None			
Examination	Written exam			
Examination duration and scale	60 min (Analysis I) + 60 min (Linear Algebra I)			
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Core Qualification: Compulsory Civil- and Environmental Engineering: Core Qualification: Compulsory Bioprocess Engineering: Core Qualification: Compulsory Digital Mechanical Engineering: Core Qualification: Compulsory Electrical Engineering: Core Qualification: Compulsory Green Technologies: Energy, Water, Climate: Core Qualification: Compulsory Computational Science and Engineering: Core Qualification: Compulsory Logistics and Mobility: Core Qualification: Compulsory Mechanical Engineering: Core Qualification: Compulsory Mechatronics: Core Qualification: Compulsory Orientation Studies: Core Qualification: Elective Compulsory Naval Architecture: Core Qualification: Compulsory Process Engineering: Core Qualification: Compulsory Engineering and Management - Major in Logistics and Mobility: Core Qualification: Compulsory			

Course L1010: Analysis I	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<p>Foundations of differential and integrational calculus of one variable</p> <ul style="list-style-type: none"> <li>• statements, sets and functions</li> <li>• natural and real numbers</li> <li>• convergence of sequences and series</li> <li>• continuous and differentiable functions</li> <li>• mean value theorems</li> <li>• Taylor series</li> <li>• calculus</li> <li>• error analysis</li> <li>• fixpoint iteration</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• <a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a></li> </ul>

Course L1012: Analysis I	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L1013: Analysis I	
<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>	Dozenten des Fachbereiches Mathematik der UHH, Dr. Simon Campese
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L0912: Linear Algebra I	
<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. Anusch Taraz, Dr. Dennis Clemens, Prof. Marko Lindner
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• vectors: intuition, rules, inner and cross product, lines and planes</li> <li>• systems of linear equations: Gauß elimination, matrix product, inverse matrices, transformations, block matrices, determinants</li> <li>• orthogonal projection in <math>\mathbb{R}^n</math>, Gram-Schmidt-Orthonormalization</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• T. Arens u.a. : Mathematik, Spektrum Akademischer Verlag, Heidelberg 2009</li> <li>• W. Mackens, H. Voß: Mathematik I für Studierende der Ingenieurwissenschaften, HECO-Verlag, Alsdorf 1994</li> <li>• W. Mackens, H. Voß: Aufgaben und Lösungen zur Mathematik I für Studierende der Ingenieurwissenschaften, HECO-Verlag, Alsdorf 1994</li> <li>• G. Strang: Lineare Algebra, Springer-Verlag, 2003</li> <li>• G. und S. Teschl: Mathematik für Informatiker, Band 1, Springer-Verlag, 2013</li> </ul>

Course L0913: Linear Algebra I	
<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. Anusch Taraz, Dr. Dennis Clemens, Prof. Marko Lindner
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>vectors: intuition, rules, inner and cross product, lines and planes</li> <li>general vector spaces: subspaces, Euclidean vector spaces</li> <li>systems of linear equations: Gauß-elimination, matrix product, inverse matrices, transformations, LR-decomposition, block matrices, determinants</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>T. Arens u.a. : Mathematik, Spektrum Akademischer Verlag, Heidelberg 2009</li> <li>W. Mackens, H. Voß: Mathematik I für Studierende der Ingenieurwissenschaften, HECO-Verlag, Alsdorf 1994</li> <li>W. Mackens, H. Voß: Aufgaben und Lösungen zur Mathematik I für Studierende der Ingenieurwissenschaften, HECO-Verlag, Alsdorf 1994</li> </ul>

Course L0914: Linear Algebra I	
<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>	Dr. Christian Seifert, Dr. Dennis Clemens
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0577: Non-technical Courses for Bachelors	
<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 Non-technical 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, migration studies, communication 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>locate selected specialized areas with the relevant non-technical mother discipline,</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>
<i>Skills</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 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 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>
<b>Personal Competence</b> <i>Social Competence</i>	<p><b>Personal Competences (Social Skills)</b></p> <p>Students will be able</p> <ul style="list-style-type: none"> <li>to learn to collaborate in different manner,</li> </ul>

## Module Manual B.Sc. "Computational Science and Engineering"

<p><i>Autonomy</i></p>	<ul style="list-style-type: none"> <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>

<p><b>Courses</b></p>	
<p><b>Information regarding lectures and courses can be found in the corresponding module handbook published separately.</b></p>	

Module M0743: Electrical Engineering I: Direct Current Networks and Electromagnetic Fields				
<b>Courses</b>				
<b>Title</b>			<b>Typ</b>	<b>Hrs/wk</b>
Electrical Engineering I: Direct Current Networks and Electromagnetic Fields (L0675)			Lecture	3
Electrical Engineering I: Direct Current Networks and Electromagnetic Fields (L0676)			Recitation Section (small)	2
<b>Module Responsible</b>	Prof. Matthias Kuhl			
<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> No	<b>Bonus</b> 10 %	<b>Form</b> Exercises	<b>Description</b>
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	120 Minutes			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Core Qualification: Compulsory Data Science: Specialisation Electrical Engineering: Compulsory Electrical Engineering: Core Qualification: Compulsory Computational Science and Engineering: Core Qualification: Compulsory Mechatronics: Core Qualification: Compulsory Orientation Studies: Core Qualification: Elective Compulsory			

Course L0675: Electrical Engineering I: Direct Current Networks and Electromagnetic Fields	
<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>	Prof. Matthias Kuhl
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	
<b>Literature</b>	<ol style="list-style-type: none"> <li>1. M. Kasper, Skript zur Vorlesung Elektrotechnik 1, 2013</li> <li>2. M. Albach: Grundlagen der Elektrotechnik 1, Pearson Education, 2004</li> <li>3. F. Moeller, H. Frohne, K.H. Löcherer, H. Müller: Grundlagen der Elektrotechnik, Teubner, 2005</li> <li>4. A. R. Hambley: Electrical Engineering, Principles and Applications, Pearson Education, 2008</li> </ol>

Course L0676: Electrical Engineering I: Direct Current Networks and Electromagnetic Fields	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 2, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Matthias Kuhl
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	
<b>Literature</b>	<ol style="list-style-type: none"> <li>1. Übungsaufgaben zur Elektrotechnik 1, TUHH, 2013</li> <li>2. Ch. Kautz: Tutorien zur Elektrotechnik, Pearson Studium, 2010</li> </ol>

Module M0547: Electrical Engineering II: Alternating Current Networks and Basic Devices				
Courses				
Title	Typ		Hrs/wk	CP
Electrical Engineering II: Alternating Current Networks and Basic Devices (L0178)	Lecture		3	5
Electrical Engineering II: Alternating Current Networks and Basic Devices (L0179)	Recitation Section (small)		2	1
<b>Module Responsible</b>	Prof. Christian Becker			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Electrical Engineering I Mathematics I Direct current networks, complex numbers			
<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 reproduce and explain fundamental theories, principles, and methods related to the theory of alternating currents. They can describe networks of linear elements using a complex notation for voltages and currents. They can reproduce an overview of applications for the theory of alternating currents in the area of electrical engineering. Students are capable of explaining the behavior of fundamental passive and active devices as well as their impact on simple circuits.</p> <p><i>Skills</i> Students are capable of calculating parameters within simple electrical networks at alternating currents by means of a complex notation for voltages and currents. They can appraise the fundamental effects that may occur within electrical networks at alternating currents. Students are able to analyze simple circuits such as oscillating circuits, filter, and matching networks quantitatively and dimension elements by means of a design. They can motivate and justify the fundamental elements of an electrical power supply (transformer, transmission line, compensation of reactive power, multiphase system) and are qualified to dimension their main features.</p> <p><i>Personal Competence</i></p> <p><i>Social Competence</i> Students are able to work together on subject related tasks in small groups. They are able to present their results effectively.</p> <p><i>Autonomy</i> Students are capable to gather necessary information from the references provided and relate that information to the context of the lecture. They are able to continually reflect their knowledge by means of activities that accompany the lecture, such as online-tests and exercises that are related to the exam. Based on respective feedback, students are expected to adjust their individual learning process. They are able to draw connections between their knowledge obtained in this lecture and the content of other lectures (e.g. Electrical Engineering I, Linear Algebra, and Analysis).</p>			
<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	10 %	Midterm	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 - 150 minutes			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Core Qualification: Compulsory Data Science: Specialisation Electrical Engineering: Compulsory Electrical Engineering: Core Qualification: Compulsory Computational Science and Engineering: Core Qualification: Compulsory Mechatronics: Core Qualification: Compulsory Orientation Studies: Core Qualification: Elective Compulsory			

Course L0178: Electrical Engineering II: Alternating Current Networks and Basic Devices	
<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>	Prof. Christian Becker
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>- General time-dependency of electrical networks</li> <li>- Representation and properties of harmonic signals</li> <li>- RLC-elements at alternating currents/voltages</li> <li>- Complex notation for the representation of RLC-elements</li> <li>- Power in electrical networks at alternating currents, compensation of reactive power</li> <li>- Frequency response locus (Nyquist plot) and Bode-diagrams</li> <li>- Measurement instrumentation for assessing alternating currents</li> <li>- Oscillating circuits, filters, electrical transmission lines</li> <li>- Transformers, three-phase current, energy converters</li> <li>- Simple non-linear and active electrical devices</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>- M. Albach, "Elektrotechnik", Pearson Studium (2011)</li> <li>- T. Harriehausen, D. Schwarzenau, "Moeller Grundlagen der Elektrotechnik", Springer (2013)</li> <li>- R. Kories, H. Schmidt-Walter, "Taschenbuch der Elektrotechnik", Harri Deutsch (2010)</li> <li>- C. Kautz, "Tutorien zur Elektrotechnik", Pearson (2009)</li> <li>- A. Hambley, "Electrical Engineering: Principles and Applications", Pearson (2013)</li> <li>- R. Dorf, "The Electrical Engineering Handbook", CRC (2006)</li> </ul>

Course L0179: Electrical Engineering II: Alternating Current Networks and Basic Devices	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 2, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Christian Becker
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>- General time-dependency of electrical networks</li> <li>- Representation and properties of harmonic signals</li> <li>- RLC-elements at alternating currents/voltages</li> <li>- Complex notation for the representation of RLC-elements</li> <li>- Power in electrical networks at alternating currents, compensation of reactive power</li> <li>- Frequency response locus (Nyquist plot) and Bode-diagrams</li> <li>- Measurement instrumentation for assessing alternating currents</li> <li>- Oscillating circuits, filters, electrical transmission lines</li> <li>- Transformers, three-phase current, energy converters</li> <li>- Simple non-linear and active electrical devices</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>- M. Albach, "Elektrotechnik", Pearson Studium (2011)</li> <li>- T. Harriehausen, D. Schwarzenau, "Moeller Grundlagen der Elektrotechnik", Springer (2013)</li> <li>- R. Kories, H. Schmidt-Walter, "Taschenbuch der Elektrotechnik", Harri Deutsch (2010)</li> <li>- C. Kautz, "Tutorien zur Elektrotechnik", Pearson (2009)</li> <li>- A. Hambley, "Electrical Engineering: Principles and Applications", Pearson (2013)</li> <li>- R. Dorf, "The Electrical Engineering Handbook", CRC (2006)</li> </ul>

Module M0624: Automata Theory and Formal Languages			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Automata Theory and Formal Languages (L0332)	Lecture	2	4
Automata Theory and Formal Languages (L0507)	Recitation Section (small)	2	2
<b>Module Responsible</b>	Prof. Matthias Mnich		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<p>Participating students should be able to</p> <ul style="list-style-type: none"> <li>- specify algorithms for simple data structures (such as, e.g., arrays) to solve computational problems</li> <li>- apply propositional logic and predicate logic for specifying and understanding mathematical proofs</li> <li>- apply the knowledge and skills taught in the module Discrete Algebraic 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>	<p>Students can explain syntax, semantics, and decision problems of propositional logic, and they are able to give algorithms for solving decision problems. Students can show correspondences to Boolean algebra. Students can describe which application problems are hard to represent with propositional logic, and therefore, the students can motivate predicate logic, and define syntax, semantics, and decision problems for this representation formalism. Students can explain unification and resolution for solving the predicate logic SAT decision problem. Students can also describe syntax, semantics, and decision problems for various kinds of temporal logic, and identify their application areas. The participants of the course can define various kinds of finite automata and can identify relationships to logic and formal grammars. The spectrum that students can explain ranges from deterministic and nondeterministic finite automata and pushdown automata to Turing machines. Students can name those formalism for which nondeterminism is more expressive than determinism. They are also able to demonstrate which decision problems require which expressivity, and, in addition, students can transform decision problems w.r.t. one formalism into decision problems w.r.t. other formalisms. They understand that some formalisms easily induce algorithms whereas others are best suited for specifying systems and their properties. Students can describe the relationships between formalisms such as logic, automata, or grammars.</p>		
<i>Skills</i>	<p>Students can apply propositional logic as well as predicate logic resolution to a given set of formulas. Students analyze application problems in order to derive propositional logic, predicate logic, or temporal logic formulas to represent them. They can evaluate which formalism is best suited for a particular application problem, and they can demonstrate the application of algorithms for decision problems to specific formulas. Students can also transform nondeterministic automata into deterministic ones, or derive grammars from automata and vice versa. They can show how parsers work, and they can apply algorithms for the language emptiness problem in case of infinite words.</p>		
<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>	90 min		
<b>Assignment for the Following Curricula</b>	<p>General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory            Computer Science: Core Qualification: Compulsory            Data Science: Core Qualification: Compulsory            Engineering Science: Specialisation Mechatronics: Elective Compulsory            General Engineering Science (English program, 7 semester): Specialisation Mechatronics: Elective Compulsory            Computational Science and Engineering: Core Qualification: Compulsory            Orientation Studies: Core Qualification: Elective Compulsory            Technomathematics: Specialisation II. Informatics: Elective Compulsory</p>		

Course L0332: Automata Theory and Formal Languages	
<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>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ol style="list-style-type: none"> <li>1. Propositional logic, Boolean algebra, propositional resolution, SAT-2KNF</li> <li>2. Predicate logic, unification, predicate logic resolution</li> <li>3. Temporal Logics (LTL, CTL)</li> <li>4. Deterministic finite automata, definition and construction</li> <li>5. Regular languages, closure properties, word problem, string matching</li> <li>6. Nondeterministic automata: <ul style="list-style-type: none"> <li>Rabin-Scott transformation of nondeterministic into deterministic automata</li> </ul> </li> <li>7. Epsilon automata, minimization of automata, elimination of e-edges, uniqueness of the minimal automaton (modulo renaming of states)</li> <li>8. Myhill-Nerode Theorem: <ul style="list-style-type: none"> <li>Correctness of the minimization procedure, equivalence classes of strings induced by automata</li> </ul> </li> <li>9. Pumping Lemma for regular languages: <ul style="list-style-type: none"> <li>provision of a tool which, in some cases, can be used to show that a finite automaton principally cannot be expressive enough to solve a word problem for some given language</li> </ul> </li> <li>10. Regular expressions vs. finite automata: <ul style="list-style-type: none"> <li>Equivalence of formalisms, systematic transformation of representations, reductions</li> </ul> </li> <li>11. Pushdown automata and context-free grammars: <ul style="list-style-type: none"> <li>Definition of pushdown automata, definition of context-free grammars, derivations, parse trees, ambiguities, pumping lemma for context-free grammars, transformation of formalisms (from pushdown automata to context-free grammars and back)</li> </ul> </li> <li>12. Chomsky normal form</li> <li>13. CYK algorithm for deciding the word problem for context-free grammars</li> <li>14. Deterministic pushdown automata</li> <li>15. Deterministic vs. nondeterministic pushdown automata: <ul style="list-style-type: none"> <li>Application for parsing, LL(k) or LR(k) grammars and parsers vs. deterministic pushdown automata, compiler compiler</li> </ul> </li> <li>16. Regular grammars</li> <li>17. Outlook: Turing machines and linear bounded automata vs general and context-sensitive grammars</li> <li>18. Chomsky hierarchy</li> <li>19. Mealy- and Moore automata: <ul style="list-style-type: none"> <li>Automata with output (w/o accepting states), infinite state sequences, automata networks</li> </ul> </li> <li>20. Omega automata: Automata for infinite input words, Büchi automata, representation of state transition systems, verification w.r.t. temporal logic specifications (in particular LTL)</li> <li>21. LTL safety conditions and model checking with Büchi automata, relationships between automata and logic</li> <li>22. Fixed points, propositional mu-calculus</li> <li>23. Characterization of regular languages by monadic second-order logic (MSO)</li> </ol>
<b>Literature</b>	<ol style="list-style-type: none"> <li>1. Logik für Informatiker Uwe Schöning, Spektrum, 5. Aufl.</li> <li>2. Logik für Informatiker Martin Kreuzer, Stefan Kühling, Pearson Studium, 2006</li> <li>3. Grundkurs Theoretische Informatik, Gottfried Vossen, Kurt-Ulrich Witt, Vieweg-Verlag, 2010.</li> <li>4. Principles of Model Checking, Christel Baier, Joost-Pieter Katoen, The MIT Press, 2007</li> </ol>

Course L0507: Automata Theory and Formal Languages	
<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>	Prof. Matthias Mnich
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0829: Foundations of Management				
Courses				
Title			Typ	Hrs/wk CP
Management Tutorial (L0882)			Recitation Section (small)	2 3
Introduction to Management (L0880)			Lecture	3 3
<b>Module Responsible</b>	Prof. Christoph Ihl			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Basic Knowledge of Mathematics and Business			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>				
<i>Knowledge</i>	<p>After taking this module, students know the important basics of many different areas in Business and Management, from Planning and Organisation to Marketing and Innovation, and also to Investment and Controlling. In particular they are able to</p> <ul style="list-style-type: none"> <li>explain the differences between Economics and Management and the sub-disciplines in Management and to name important definitions from the field of Management</li> <li>explain the most important aspects of and goals in Management and name the most important aspects of entrepreneurial projects</li> <li>describe and explain basic business functions as production, procurement and sourcing, supply chain management, organization and human resource management, information management, innovation management and marketing</li> <li>explain the relevance of planning and decision making in Business, esp. in situations under multiple objectives and uncertainty, and explain some basic methods from mathematical Finance</li> <li>state basics from accounting and costing and selected controlling methods.</li> </ul>			
<i>Skills</i>	<p>Students are able to analyse business units with respect to different criteria (organization, objectives, strategies etc.) and to carry out an Entrepreneurship project in a team. In particular, they are able to</p> <ul style="list-style-type: none"> <li>analyse Management goals and structure them appropriately</li> <li>analyse organisational and staff structures of companies</li> <li>apply methods for decision making under multiple objectives, under uncertainty and under risk</li> <li>analyse production and procurement systems and Business information systems</li> <li>analyse and apply basic methods of marketing</li> <li>select and apply basic methods from mathematical finance to predefined problems</li> <li>apply basic methods from accounting, costing and controlling to predefined problems</li> </ul>			
<b>Personal Competence</b>				
<i>Social Competence</i>	<p>Students are able to</p> <ul style="list-style-type: none"> <li>work successfully in a team of students</li> <li>to apply their knowledge from the lecture to an entrepreneurship project and write a coherent report on the project</li> <li>to communicate appropriately and</li> <li>to cooperate respectfully with their fellow students.</li> </ul>			
<i>Autonomy</i>	<p>Students are able to</p> <ul style="list-style-type: none"> <li>work in a team and to organize the team themselves</li> <li>to write a report on their project.</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>	Subject theoretical and practical work			
<b>Examination duration and scale</b>	several written exams during the semester			
<b>Assignment for the Following Curricula</b>	<p>General Engineering Science (German program, 7 semester): Core Qualification: Compulsory            Civil- and Environmental Engineering: Specialisation Civil Engineering: Elective Compulsory            Civil- and Environmental Engineering: Specialisation Water and Environment: Elective Compulsory            Civil- and Environmental Engineering: Specialisation Traffic and Mobility: Elective Compulsory            Bioprocess Engineering: Core Qualification: Compulsory            Computer Science: Core Qualification: Compulsory            Data Science: Core Qualification: Compulsory            Electrical Engineering: Core Qualification: Compulsory            Energy and Environmental Engineering: Core Qualification: Compulsory            General Engineering Science (English program, 7 semester): Specialisation Electrical Engineering: Compulsory            General Engineering Science (English program, 7 semester): Specialisation Civil Engineering: Compulsory            General Engineering Science (English program, 7 semester): Specialisation Bioprocess Engineering: Compulsory            General Engineering Science (English program, 7 semester): Specialisation Energy and Environmental Engineering: Compulsory            General Engineering Science (English program, 7 semester): Specialisation Computer Science: Compulsory            General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Biomechanics: Compulsory            General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Energy Systems: Compulsory            General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Aircraft Systems Engineering: Compulsory</p>			

## Module Manual B.Sc. "Computational Science and Engineering"

	<p>General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Materials in Engineering Sciences: Compulsory</p> <p>General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Mechatronics: Compulsory</p> <p>General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Product Development and Production: Compulsory</p> <p>General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Theoretical Mechanical Engineering: Compulsory</p> <p>General Engineering Science (English program, 7 semester): Specialisation Naval Architecture: Compulsory</p> <p>General Engineering Science (English program, 7 semester): Specialisation Process Engineering: Compulsory</p> <p>General Engineering Science (English program, 7 semester): Specialisation Biomedical Engineering: Compulsory</p> <p>Green Technologies: Energy, Water, Climate: Core Qualification: Compulsory</p> <p>Computational Science and Engineering: Core Qualification: Compulsory</p> <p>Logistics and Mobility: Core Qualification: Compulsory</p> <p>Mechanical Engineering: Core Qualification: Compulsory</p> <p>Mechatronics: Core Qualification: Compulsory</p> <p>Orientation Studies: Core Qualification: Elective Compulsory</p> <p>Orientation Studies: Core Qualification: Elective Compulsory</p> <p>Naval Architecture: Core Qualification: Compulsory</p> <p>Technomathematics: Core Qualification: Compulsory</p> <p>Process Engineering: Core Qualification: Compulsory</p> <p>Engineering and Management - Major in Logistics and Mobility: Core Qualification: Compulsory</p>
--	--

Course L0882: Management Tutorial	
<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. Christoph Ihl, Katharina Roedelius
<b>Language</b>	DE
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	<p>In the management tutorial, the contents of the lecture will be deepened by practical examples and the application of the discussed tools.</p> <p>If there is adequate demand, a problem-oriented tutorial will be offered in parallel, which students can choose alternatively. Here, students work in groups on selected projects that focus on the elaboration of an innovative business idea from the point of view of an established company or a startup. Again, the business knowledge from the lecture should come to practical use. The group projects are guided by a mentor.</p>
<b>Literature</b>	Relevante Literatur aus der korrespondierenden Vorlesung.

# Module Manual B.Sc. "Computational Science and Engineering"

Course L0880: Introduction to Management	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 48, Study Time in Lecture 42
<b>Lecturer</b>	Prof. Christoph Ihl, Prof. Christian Lüthje, Prof. Christian Ringle, Prof. Cornelius Herstatt, Prof. Kathrin Fischer, Prof. Matthias Meyer, Prof. Thomas Wrona, Prof. Thorsten Blecker, Prof. Wolfgang Kersten
<b>Language</b>	DE
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction to Business and Management, Business versus Economics, relevant areas in Business and Management</li> <li>• Important definitions from Management,</li> <li>• Developing Objectives for Business, and their relation to important Business functions</li> <li>• Business Functions: Functions of the Value Chain, e.g. Production and Procurement, Supply Chain Management, Innovation Management, Marketing and Sales</li> <li>• Cross-sectional Functions, e.g. Organisation, Human Ressource Management, Supply Chain Management, Information Management</li> <li>• Definitions as information, information systems, aspects of data security and strategic information systems</li> <li>• Definition and Relevance of innovations, e.g. innovation opportunities, risks etc.</li> <li>• Relevance of marketing, B2B vs. B2C-Marketing</li> <li>• different techniques from the field of marketing (e.g. scenario technique), pricing strategies</li> <li>• important organizational structures</li> <li>• basics of human ressource management</li> <li>• Introduction to Business Planning and the steps of a planning process</li> <li>• Decision Analysis: Elements of decision problems and methods for solving decision problems</li> <li>• Selected Planning Tasks, e.g. Investment and Financial Decisions</li> <li>• Introduction to Accounting: Accounting, Balance-Sheets, Costing</li> <li>• Relevance of Controlling and selected Controlling methods</li> <li>• Important aspects of Entrepreneurship projects</li> </ul>
<b>Literature</b>	<p>Bamberg, G., Coenenberg, A.: Betriebswirtschaftliche Entscheidungslehre, 14. Aufl., München 2008</p> <p>Eisenführ, F., Weber, M.: Rationales Entscheiden, 4. Aufl., Berlin et al. 2003</p> <p>Heinhold, M.: Buchführung in Fallbeispielen, 10. Aufl., Stuttgart 2006.</p> <p>Kruschwitz, L.: Finanzmathematik. 3. Auflage, München 2001.</p> <p>Pellens, B., Fülber, R. U., Gassen, J., Sellhorn, T.: Internationale Rechnungslegung, 7. Aufl., Stuttgart 2008.</p> <p>Schweitzer, M.: Planung und Steuerung, in: Bea/Friedl/Schweitzer: Allgemeine Betriebswirtschaftslehre, Bd. 2: Führung, 9. Aufl., Stuttgart 2005.</p> <p>Weber, J., Schäffer, U. : Einführung in das Controlling, 12. Auflage, Stuttgart 2008.</p> <p>Weber, J./Weißenberger, B.: Einführung in das Rechnungswesen, 7. Auflage, Stuttgart 2006.</p>

Module M0851: Mathematics II				
Courses				
Title		Typ	Hrs/wk	CP
Analysis II (L1025)		Lecture	2	2
Analysis II (L1026)		Recitation Section (large)	1	1
Analysis II (L1027)		Recitation Section (small)	1	1
Linear Algebra II (L0915)		Lecture	2	2
Linear Algebra II (L0916)		Recitation Section (small)	1	1
Linear Algebra II (L0917)		Recitation Section (large)	1	1
Module Responsible	Prof. Anusch Taraz			
Admission Requirements	None			
Recommended Previous Knowledge	Mathematics I			
Educational Objectives	After taking part successfully, students have reached the following learning results			
Professional Competence	<div>Knowledge</div> <ul style="list-style-type: none"><li>Students can name further concepts in analysis and linear algebra. 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> <div>Skills</div> <ul style="list-style-type: none"><li>Students can model problems in analysis and linear algebra 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> <div>Personal Competence</div> <div>Social Competence</div> <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> <div>Autonomy</div> <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>			
Workload in Hours	Independent Study Time 128, Study Time in Lecture 112			
Credit points	8			
Course achievement	None			
Examination	Written exam			
Examination duration and scale	60 min (Analysis II) + 60 min (Linear Algebra II)			
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Core Qualification: Compulsory Civil- and Environmental Engineering: Core Qualification: Compulsory Bioprocess Engineering: Core Qualification: Compulsory Digital Mechanical Engineering: Core Qualification: Compulsory Electrical Engineering: Core Qualification: Compulsory Green Technologies: Energy, Water, Climate: Core Qualification: Compulsory Computational Science and Engineering: Core Qualification: Compulsory Logistics and Mobility: Core Qualification: Compulsory Mechanical Engineering: Core Qualification: Compulsory Mechatronics: Core Qualification: Compulsory Orientation Studies: Core Qualification: Elective Compulsory Naval Architecture: Core Qualification: Compulsory Process Engineering: Core Qualification: Compulsory Engineering and Management - Major in Logistics and Mobility: Core Qualification: Compulsory			

Course L1025: Analysis II	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• power series and elementary functions</li> <li>• interpolation</li> <li>• integration (proper integrals, fundamental theorem, integration rules, improper integrals, parameter dependent integrals)</li> <li>• applications of integration (volume and surface of bodies of revolution, lines and arc length, line integrals)</li> <li>• numerical quadrature</li> <li>• periodic functions</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• <a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a></li> </ul>

Course L1026: Analysis II	
<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>	Dozenten des Fachbereiches Mathematik der UHH, Dr. Sebastian Götschel
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L1027: Analysis II	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L0915: Linear Algebra II	
<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. Anusch Taraz, Dr. Dennis Clemens, Prof. Marko Lindner
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• general vector spaces: subspaces, Euclidean vector spaces</li> <li>• linear mappings: basis transformation, orthogonal projection, orthogonal matrices, householder matrices</li> <li>• linear regression: normal equations, linear discrete approximation</li> <li>• eigenvalues: diagonalising matrices, normal matrices, symmetric and Hermite matrices</li> <li>• system of linear differential equations</li> <li>• matrix factorizations: LR-decomposition, QR-decomposition, Schur decomposition, Jordan normal form, singular value decomposition</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• T. Arens u.a. : Mathematik, Spektrum Akademischer Verlag, Heidelberg 2009</li> <li>• W. Mackens, H. Voß: Mathematik I für Studierende der Ingenieurwissenschaften, HECO-Verlag, Alsdorf 1994</li> <li>• W. Mackens, H. Voß: Aufgaben und Lösungen zur Mathematik I für Studierende der Ingenieurwissenschaften, HECO-Verlag, Alsdorf 1994</li> <li>• G. Strang: Lineare Algebra, Springer-Verlag, 2003</li> <li>• G. und S. Teschl: Mathematik für Informatiker, Band 1, Springer-Verlag, 2013</li> </ul>

Course L0916: Linear Algebra II	
<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. Anusch Taraz, Dr. Dennis Clemens, Prof. Marko Lindner
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>linear mappings: basis transformation, orthogonal projection, orthogonal matrices, householder matrices</li> <li>linear regression: QR-decomposition, normal equations, linear discrete approximation</li> <li>eigenvalues: diagonalising matrices, normal matrices, symmetric and Hermite matrices, Jordan normal form, singular value decomposition</li> <li>system of linear differential equations</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>W. Mackens, H. Voß: Mathematik I für Studierende der Ingenieurwissenschaften, HECO-Verlag, Alsdorf 1994</li> <li>W. Mackens, H. Voß: Aufgaben und Lösungen zur Mathematik I für Studierende der Ingenieurwissenschaften, HECO-Verlag, Alsdorf 1994</li> </ul>

Course L0917: Linear Algebra II	
<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. Anusch Taraz, Dr. Christian Seifert, Dr. Dennis Clemens, Prof. Marko Lindner
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1432: Programming Paradigms				
Courses				
Title		Typ	Hrs/wk	CP
Programming Paradigms (L2169)		Lecture	2	2
Programming Paradigms (L2170)		Recitation Section (large)	1	1
Programming Paradigms (L2171)		Practical Course	2	3
Module Responsible	Dr. Thibaut Lunet			
Admission Requirements	None			
Recommended Previous Knowledge	Lecture on procedural programming or equivalent programming skills			
Educational Objectives	After taking part successfully, students have reached the following learning results			
Professional Competence	<p><i>Knowledge</i> The students have a fundamental understanding of object orientated and generic programming and can apply it in small programming projects. The can design own class hierarchies and differentiate between different ways of inheritance. They have a fundamental understanding of polymorphism and can differentiate between run-time and compile-time polymorphism. The students know the concept of information hiding and can design interfaces with public and private methods. They can use exceptions and apply generic programming in order to make existing data structures generic. The students know the pros and cons of both programming paradigms.</p> <p><i>Skills</i> Students can break down a medium-sized problem into subproblems and create their own classes in an object-oriented programming language based on these subproblems. They can design a public and private interface and implement the implementation generically and extensible by abstraction. They can distinguish different language constructs of a modern programming language and use these suitably in the implementation. They can design and implement unit tests.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> Students can work in teams and communicate in forums.</p> <p><i>Autonomy</i> In a programming internship, students learn object-oriented programming under supervision. In exercises they develop individual and independent solutions and receive feedback.</p>			
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70			
Credit points	6			
Course achievement	None			
Examination	Written exam			
Examination duration and scale	90 min			
Assignment for the Following Curricula	Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Computational Science and Engineering: Core Qualification: Compulsory Technomathematics: Core Qualification: Compulsory			

Course L2169: Programming Paradigms	
<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>	Dr. Thibaut Lunet
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• fundamentals behind object orientated programming</li> <li>• classes and objects</li> <li>• inheritance (single, multiple)</li> <li>• interfaces</li> <li>• information hiding</li> <li>• exception handling</li> <li>• generic programming and the implementation in the compiler</li> <li>• excursus in programming with dynamically typed programming languages</li> </ul>
<b>Literature</b>	Skript

Course L2170: Programming Paradigms	
<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>	Dr. Thibaut Lunet
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• fundamentals behind object orientated programming</li> <li>• classes and objects</li> <li>• inheritance (single, multiple)</li> <li>• interfaces</li> <li>• information hiding</li> <li>• exception handling</li> <li>• generic programming and the implementation in the compiler</li> <li>• excursus in programming with dynamically typed programming languages</li> </ul>
<b>Literature</b>	Skript

Course L2171: Programming Paradigms	
<b>Typ</b>	Practical Course
<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. Thibaut Lunet
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• fundamentals behind object orientated programming</li> <li>• classes and objects</li> <li>• inheritance (single, multiple)</li> <li>• interfaces</li> <li>• information hiding</li> <li>• exception handling</li> <li>• generic programming and the implementation in the compiler</li> <li>• excursus in programming with dynamically typed programming languages</li> </ul>
<b>Literature</b>	Skript

Module M0834: Computernetworks and Internet Security			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Computer Networks and Internet Security (L1098)	Lecture	3	5
Computer Networks and Internet Security (L1099)	Recitation Section (small)	1	1
<b>Module Responsible</b>	Prof. Andreas Timm-Giel		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Basics of Computer Science		
<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 explain important and common Internet protocols in detail and classify them, in order to be able to analyse and develop networked systems in further studies and job.</p> <p><i>Skills</i> Students are able to analyse common Internet protocols and evaluate the use of them in different domains.</p> <p><i>Personal Competence</i></p> <p><i>Social Competence</i></p> <p><i>Autonomy</i> Students can select relevant parts out of high amount of professional knowledge and can independently learn and understand 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>	120 min		
<b>Assignment for the Following Curricula</b>	<p>General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory</p> <p>Computer Science: Core Qualification: Compulsory</p> <p>Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory</p> <p>Data Science: Core Qualification: Elective Compulsory</p> <p>Electrical Engineering: Core Qualification: Elective Compulsory</p> <p>Engineering Science: Specialisation Electrical Engineering: Elective Compulsory</p> <p>Engineering Science: Specialisation Mechatronics: Elective Compulsory</p> <p>Engineering Science: Specialisation Mechatronics: Elective Compulsory</p> <p>General Engineering Science (English program, 7 semester): Specialisation Mechatronics: Elective Compulsory</p> <p>Computer Science in Engineering: Core Qualification: Compulsory</p> <p>Technomathematics: Specialisation II. Informatics: Elective Compulsory</p>		

Course L1098: Computer Networks and Internet Security	
<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>	Prof. Andreas Timm-Giel, Dr.-Ing. Koojana Kuladinithi, Prof. Dieter Gollmann
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<p>In this class an introduction to computer networks with focus on the Internet and its security is given. Basic functionality of complex protocols are introduced. Students learn to understand these and identify common principles. In the exercises these basic principles and an introduction to performance modelling are addressed using computing tasks and (virtual) labs.</p> <p>In the second part of the lecture an introduction to Internet security is given.</p> <p>This class comprises:</p> <ul style="list-style-type: none"> <li>• Application layer protocols (HTTP, FTP, DNS)</li> <li>• Transport layer protocols (TCP, UDP)</li> <li>• Network Layer (Internet Protocol, routing in the Internet)</li> <li>• Data link layer with media access at the example of Ethernet</li> <li>• Multimedia applications in the Internet</li> <li>• Network management</li> <li>• Internet security: IPSec</li> <li>• Internet security: Firewalls</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Kurose, Ross, Computer Networking - A Top-Down Approach, 6th Edition, Addison-Wesley</li> <li>• Kurose, Ross, Computernetzwerke - Der Top-Down-Ansatz, Pearson Studium; Auflage: 6. Auflage</li> <li>• W. Stallings: Cryptography and Network Security: Principles and Practice, 6th edition</li> </ul> <p>Further literature is announced at the beginning of the lecture.</p>

Course L1099: Computer Networks and Internet Security	
<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. Andreas Timm-Giel, Prof. Dieter Gollmann
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0662: Numerical Mathematics I				
Courses				
Title	Typ		Hrs/wk	CP
Numerical Mathematics I (L0417)	Lecture		2	3
Numerical Mathematics I (L0418)	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>Mathematik I + II for Engineering Students (german or english) <b>or</b> Analysis &amp; Linear Algebra I + II for Technomathematicians</li> <li>basic MATLAB/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 numerical methods for interpolation, integration, least squares problems, eigenvalue problems, nonlinear root finding problems and to explain their core ideas,</li> <li>repeat convergence statements for the numerical methods,</li> <li>explain aspects for the practical execution of numerical methods with respect to computational and storage complexitx.</li> </ul>			
<i>Skills</i>	<p>Students are able to</p> <ul style="list-style-type: none"> <li>implement, apply and compare numerical methods using MATLAB/Python,</li> <li>justify the convergence behaviour of numerical methods with respect to the problem and solution algorithm,</li> <li>select and execute a suitable solution approach for a given problem.</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 excercises 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 minutes			
<b>Assignment for the Following Curricula</b>	<p>General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Biomedical Engineering: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Biomechanics: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Theoretical Mechanical Engineering: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Aircraft Systems Engineering: Elective Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Mechatronics: Elective Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Energy Systems: Elective Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Advanced Materials: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Materials in Engineering Sciences: Compulsory</p> <p>Bioprocess Engineering: Specialisation A - General Bioprocess Engineering: Elective Compulsory</p> <p>Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory</p> <p>Data Science: Core Qualification: Compulsory</p> <p>Electrical Engineering: Core Qualification: Elective Compulsory</p> <p>Engineering Science: Core Qualification: Compulsory</p> <p>Engineering Science: Core Qualification: Compulsory</p> <p>Computer Science in Engineering: Core Qualification: Compulsory</p> <p>Mechanical Engineering: Specialisation Theoretical Mechanical Engineering: Compulsory</p> <p>Mechanical Engineering: Specialisation Energy Systems: Elective Compulsory</p> <p>Theoretical Mechanical Engineering: Technical Complementary Course Core Studies: Elective Compulsory</p> <p>Process Engineering: Specialisation Process Engineering: Elective Compulsory</p>			

Course L0417: Numerical Mathematics I	
<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>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ol style="list-style-type: none"> <li>1. Finite precision arithmetic, error analysis, conditioning and stability</li> <li>2. Linear systems of equations: LU and Cholesky factorization, condition</li> <li>3. Interpolation: polynomial, spline and trigonometric interpolation</li> <li>4. Nonlinear equations: fixed point iteration, root finding algorithms, Newton's method</li> <li>5. Linear and nonlinear least squares problems: normal equations, Gram Schmidt and Householder orthogonalization, singular value decomposition, regularization, Gauss-Newton and Levenberg-Marquardt methods</li> <li>6. Eigenvalue problems: power iteration, inverse iteration, QR algorithm</li> <li>7. Numerical differentiation</li> <li>8. Numerical integration: Newton-Cotes rules, error estimates, Gauss quadrature, adaptive quadrature</li> </ol>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Gander/Gander/Kwok: Scientific Computing: An introduction using Maple and MATLAB, Springer (2014)</li> <li>• Stoer/Bulirsch: Numerische Mathematik 1, Springer</li> <li>• Dahmen, Reusken: Numerik für Ingenieure und Naturwissenschaftler, Springer</li> </ul>

Course L0418: Numerical Mathematics I	
<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>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0730: Computer Engineering				
Courses				
Title	Typ		Hrs/wk	CP
Computer Engineering (L0321)	Lecture		3	4
Computer Engineering (L0324)	Recitation Section (small)		1	2
<b>Module Responsible</b>	Prof. Heiko Falk			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Basic knowledge in electrical engineering			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b> <i>Knowledge</i>	<p>This module deals with the foundations of the functionality of computing systems. It covers the layers from the assembly-level programming down to gates. The module includes the following topics:</p> <ul style="list-style-type: none"> <li>• Introduction</li> <li>• Combinational logic: Gates, Boolean algebra, Boolean functions, hardware synthesis, combinational networks</li> <li>• Sequential logic: Flip-flops, automata, systematic hardware design</li> <li>• Technological foundations</li> <li>• Computer arithmetic: Integer addition, subtraction, multiplication and division</li> <li>• Basics of computer architecture: Programming models, MIPS single-cycle architecture, pipelining</li> <li>• Memories: Memory hierarchies, SRAM, DRAM, caches</li> <li>• Input/output: I/O from the perspective of the CPU, principles of passing data, point-to-point connections, busses</li> </ul>			
<i>Skills</i>	<p>The students perceive computer systems from the architect's perspective, i.e., they identify the internal structure and the physical composition of computer systems. The students can analyze, how highly specific and individual computers can be built based on a collection of few and simple components. They are able to distinguish between and to explain the different abstraction layers of today's computing systems - from gates and circuits up to complete processors.</p> <p>After successful completion of the module, the students are able to judge the interdependencies between a physical computer system and the software executed on it. In particular, they shall understand the consequences that the execution of software has on the hardware-centric abstraction layers from the assembly language down to gates. This way, they will be enabled to evaluate the impact that these low abstraction levels have on an entire system's performance and to propose feasible options.</p>			
<b>Personal Competence</b> <i>Social Competence</i>	Students are able to solve similar problems alone or in a group and to present the results accordingly.			
<i>Autonomy</i>	Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.			
<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	10 %	Exercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 minutes, contents of course and labs			
<b>Assignment for the Following Curricula</b>	<p>General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Mechatronics: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Aircraft Systems Engineering: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Theoretical Mechanical Engineering: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Materials in Engineering Sciences: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Product Development and Production: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Energy Systems: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Biomechanics: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Electrical Engineering: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Green Technologies, Focus Renewable Energy: Elective Compulsory</p> <p>Computer Science: Core Qualification: Compulsory</p> <p>Data Science: Core Qualification: Elective Compulsory</p> <p>Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory</p> <p>Electrical Engineering: Core Qualification: Compulsory</p> <p>Computer Science in Engineering: Core Qualification: Compulsory</p> <p>Integrated Building Technology: Core Qualification: Elective Compulsory</p> <p>Technomathematics: Specialisation II. Informatics: Elective Compulsory</p>			

Course L0321: Computer Engineering	
<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>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Combinational Logic</li> <li>• Sequential Logic</li> <li>• Technological Foundations</li> <li>• Representations of Numbers, Computer Arithmetics</li> <li>• Foundations of Computer Architecture</li> <li>• Memories</li> <li>• Input/Output</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• A. Clements. The Principles of Computer Hardware. 3. Auflage, Oxford University Press, 2000.</li> <li>• A. Tanenbaum, J. Goodman. Computerarchitektur. Pearson, 2001.</li> <li>• D. Patterson, J. Hennessy. Rechnerorganisation und -entwurf. Elsevier, 2005.</li> </ul>

Course L0324: Computer Engineering	
<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. Heiko Falk
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0853: Mathematics III				
Courses				
Title		Typ	Hrs/wk	CP
Analysis III (L1028)		Lecture	2	2
Analysis III (L1029)		Recitation Section (small)	1	1
Analysis III (L1030)		Recitation Section (large)	1	1
Differential Equations 1 (Ordinary Differential Equations) (L1031)		Lecture	2	2
Differential Equations 1 (Ordinary Differential Equations) (L1032)		Recitation Section (small)	1	1
Differential Equations 1 (Ordinary Differential Equations) (L1033)		Recitation Section (large)	1	1
Module Responsible	Prof. Anusch Taraz			
Admission Requirements	None			
Recommended Previous Knowledge	Mathematics I + II			
Educational Objectives	After taking part successfully, students have reached the following learning results			
Professional Competence	<div>Knowledge</div> <ul style="list-style-type: none"><li>Students can name the basic concepts in the area of analysis and differential equations. 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> <div>Skills</div> <ul style="list-style-type: none"><li>Students can model problems in the area of analysis and differential equations 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> <div>Personal Competence</div> <div>Social Competence</div> <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> <div>Autonomy</div> <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>			
Workload in Hours	Independent Study Time 128, Study Time in Lecture 112			
Credit points	8			
Course achievement	None			
Examination	Written exam			
Examination duration and scale	60 min (Analysis III) + 60 min (Differential Equations 1)			
Assignment for the Following Curricula	<div>General Engineering Science (German program, 7 semester): Core Qualification: Compulsory</div> <div>Civil- and Environmental Engineering: Core Qualification: Compulsory</div> <div>Bioprocess Engineering: Core Qualification: Compulsory</div> <div>Chemical and Bioprocess Engineering: Core Qualification: Compulsory</div> <div>Digital Mechanical Engineering: Core Qualification: Compulsory</div> <div>Electrical Engineering: Core Qualification: Compulsory</div> <div>Green Technologies: Energy, Water, Climate: Core Qualification: Compulsory</div> <div>Computer Science in Engineering: Core Qualification: Compulsory</div> <div>Integrated Building Technology: Core Qualification: Compulsory</div> <div>Logistics and Mobility: Specialisation Traffic Planning and Systems: Elective Compulsory</div> <div>Logistics and Mobility: Specialisation Production Management and Processes: Elective Compulsory</div> <div>Logistics and Mobility: Specialisation Information Technology: Compulsory</div> <div>Mechanical Engineering: Core Qualification: Compulsory</div> <div>Mechatronics: Core Qualification: Compulsory</div> <div>Naval Architecture: Core Qualification: Compulsory</div> <div>Process Engineering: Core Qualification: Compulsory</div> <div>Engineering and Management - Major in Logistics and Mobility: Specialisation Traffic Planning and Systems: Elective Compulsory</div> <div>Engineering and Management - Major in Logistics and Mobility: Specialisation Production Management and Processes: Elective Compulsory</div> <div>Engineering and Management - Major in Logistics and Mobility: Specialisation Information Technology: Compulsory</div>			

Course L1028: Analysis III	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<p>Main features of differential and integrational calculus of several variables</p> <ul style="list-style-type: none"> <li>• Differential calculus for several variables</li> <li>• Mean value theorems and Taylor's theorem</li> <li>• Maximum and minimum values</li> <li>• Implicit functions</li> <li>• Minimization under equality constraints</li> <li>• Newton's method for multiple variables</li> <li>• Double integrals over general regions</li> <li>• Line and surface integrals</li> <li>• Theorems of Gauß and Stokes</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• <a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a></li> </ul>

Course L1029: Analysis III	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L1030: Analysis III	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L1031: Differential Equations 1 (Ordinary Differential Equations)	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<p>Main features of the theory and numerical treatment of ordinary differential equations</p> <ul style="list-style-type: none"> <li>• Introduction and elementary methods</li> <li>• Existence and uniqueness of initial value problems</li> <li>• Linear differential equations</li> <li>• Stability and qualitative behaviour of the solution</li> <li>• Boundary value problems and basic concepts of calculus of variations</li> <li>• Eigenvalue problems</li> <li>• Numerical methods for the integration of initial and boundary value problems</li> <li>• Classification of partial differential equations</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• <a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a></li> </ul>

Course L1032: Differential Equations 1 (Ordinary Differential Equations)	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L1033: Differential Equations 1 (Ordinary Differential Equations)	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1423: Algorithms and Data Structures			
Courses			
Title	Typ	Hrs/wk	CP
Algorithms and Data Structures (L2046)	Lecture	4	4
Algorithms and Data Structures (L2047)	Recitation Section (small)	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>Mathematics II</li> <li>Procedural Programming</li> <li>Objectoriented Programming</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 algorithm design, algorithm analysis and problem reductions. 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>Skills</i> <ul style="list-style-type: none"> <li>Students can model discrete decision, search and optimization problems with the help of the concepts studied in this course. Moreover, they are capable of solving them, and reducing them to each other, 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 110, Study Time in Lecture 70		
<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>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Logistics and Mobility: Specialisation Information Technology: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory Engineering and Management - Major in Logistics and Mobility: Specialisation Information Technology: Elective Compulsory		

Course L2046: Algorithms and Data Structures	
<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>• Insertion sort</li> <li>• Register machines</li> <li>• Asymptotic analysis, Landau notation</li> <li>• Polynomial-time algorithms and NP-completeness</li> <li>• Divide-and-conquer, merge sort</li> <li>• Strassen algorithm</li> <li>• Greedy algorithm</li> <li>• Dynamic programming</li> <li>• Quick sort</li> <li>• AVL-trees, B-trees</li> <li>• Hashing</li> <li>• Depth first search, breadth first search</li> <li>• Shortest paths</li> <li>• Flow problems, Ford-Fulkerson algorithm</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• T. Cormen, Ch. Leiserson, R. Rivest, C. Stein: Introduction to Algorithms. MIT Press, 2013</li> <li>• S. Skiena: The Algorithm Design Manual. Springer, 2008</li> <li>• J. M. Kleinberg and É. Tardos. Algorithm Design. Addison-Wesley, 2005.</li> </ul>

Course L2047: Algorithms and Data Structures	
<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 Mnich
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1578: Seminars Computer Science			
Courses			
Title	Typ	Hrs/wk	CP
Introductory Seminar Computer Science I (L2362)	Seminar	2	3
Introductory Seminar Computer Science II (L2361)	Seminar	2	3
Module Responsible	Dozenten des SD E		
Admission Requirements	None		
Recommended Previous Knowledge	Basic knowledge of Computer Science and Mathematics at the Bachelor's level.		
Educational Objectives	After taking part successfully, students have reached the following learning results		
Professional Competence	<div><div>Knowledge</div><div>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></div><div><div>Skills</div><div>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></div></div></div>		
Personal Competence			
Social Competence			
Autonomy			
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56		
Credit points	6		
Course achievement	None		
Examination	Presentation		
Examination duration and scale	x		
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory		

Course L2362: Introductory 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>	Dozenten des SD E
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	
<b>Literature</b>	

Course L2361: 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>	Dozenten des SD E
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	
<b>Literature</b>	

Module M0672: Signals and Systems				
Courses				
Title	Typ		Hrs/wk	CP
Signals and Systems (L0432)	Lecture		3	4
Signals and Systems (L0433)	Recitation Section (small)		2	2
<b>Module Responsible</b>	Prof. Gerhard Bauch			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	<p>Mathematics 1-3</p> <p>The modul is an introduction to the theory of signals and systems. Good knowledge in maths as covered by the moduls Mathematik 1-3 is expected. Further experience with spectral transformations (Fourier series, Fourier transform, Laplace transform) is useful but not required.</p>			
<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 are able to classify and describe signals and linear time-invariant (LTI) systems using methods of signal and system theory. They are able to apply the fundamental transformations of continuous-time and discrete-time signals and systems. They can describe and analyse deterministic signals and systems mathematically in both time and image domain. In particular, they understand the effects in time domain and image domain which are caused by the transition of a continuous-time signal to a discrete-time signal.</p> <p>The students are familiar with the contents of lecture and tutorials. They can explain and apply them to new problems.</p> <p><i>Skills</i> The students are able to describe and analyse deterministic signals and linear time-invariant systems using methods of signal and system theory. They can analyse and design basic systems regarding important properties such as magnitude and phase response, stability, linearity etc.. They can assess the impact of LTI systems on the signal properties in time and frequency domain.</p>			
<b>Personal Competence</b>	<p><i>Social Competence</i> The students can jointly solve specific problems.</p> <p><i>Autonomy</i> The students are able to acquire relevant information from appropriate literature sources. They can control their level of knowledge during the lecture period by solving tutorial problems, software tools, clicker system.</p>			
<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>	Written exam			
<b>Examination duration and scale</b>	90 min			
<b>Assignment for the Following Curricula</b>	<p>General Engineering Science (German program, 7 semester): Core Qualification: Compulsory</p> <p>Computer Science: Core Qualification: Compulsory</p> <p>Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory</p> <p>Data Science: Core Qualification: Compulsory</p> <p>Electrical Engineering: Core Qualification: Compulsory</p> <p>Computer Science in Engineering: Core Qualification: Compulsory</p> <p>Integrated Building Technology: Core Qualification: Compulsory</p> <p>Mechatronics: Core Qualification: Compulsory</p> <p>Technomathematics: Specialisation III. Engineering Science: Elective Compulsory</p>			

Course L0432: Signals and 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. Gerhard Bauch, Dr. Rainer Grünheid
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction to signal and system theory</li> <li>• Signals <ul style="list-style-type: none"> <li>◦ Classification of signals <ul style="list-style-type: none"> <li>■ Continuous-time and discrete-time signals</li> <li>■ Analog and digital signals</li> <li>■ Deterministic and random signals</li> </ul> </li> <li>◦ Description of LTI systems by differential equations or difference equations, respectively</li> <li>◦ Basic properties of signals and operations on signals</li> <li>◦ Elementary signals</li> <li>◦ Distributions (Generalized Functions)</li> <li>◦ Power and energy of signals</li> <li>◦ Correlation functions of deterministic signals <ul style="list-style-type: none"> <li>■ Autocorrelation function</li> <li>■ Crosscorrelation function</li> <li>■ Orthogonal signals</li> <li>■ Applications of correlation</li> </ul> </li> </ul> </li> <li>• Linear time-invariant (LTI) systems</li> </ul>

	<ul style="list-style-type: none"> <li>◦ Linearity</li> <li>◦ Time-invariance</li> <li>◦ Description of LTI systems by impulse response and frequency response</li> <li>◦ Convolution</li> <li>◦ Convolution and correlation</li> <li>◦ Properties of LTI-systems</li> <li>◦ Causal systems</li> <li>◦ Stable systems</li> <li>◦ Memoryless systems</li> <li>• Fourier Series and Fourier Transform <ul style="list-style-type: none"> <li>◦ Fourier transform of continuous-time signals, discrete-time signals, periodic signals, non-periodic signals</li> <li>◦ Properties of the Fourier transform</li> <li>◦ Fourier transform of some basic signals</li> <li>◦ Parseval's theorem</li> </ul> </li> <li>• Analysis of LTI-systems and signals in the frequency domain <ul style="list-style-type: none"> <li>◦ Frequency response, magnitude response and phase response</li> <li>◦ Transmission factor, attenuation, gain</li> <li>◦ Frequency-flat and frequency-selective LTI-systems</li> <li>◦ Bandwidth definitions</li> <li>◦ Basic types of systems (filters), lowpass, highpass, bandpass, bandstop systems</li> <li>◦ Phase delay and group delay</li> <li>◦ Linear-phase systems</li> <li>◦ Distortion-free systems</li> <li>◦ Spectrum analysis with limited observation window: Leakage effect</li> </ul> </li> <li>• Laplace Transform <ul style="list-style-type: none"> <li>◦ Relation of Fourier transform and Laplace transform</li> <li>◦ Properties of the Laplace transform</li> <li>◦ Laplace transform of some basic signals</li> </ul> </li> <li>• Analysis of LTI-systems in the s-domain <ul style="list-style-type: none"> <li>◦ Transfer function of LTI-systems</li> <li>◦ Relation of Laplace transform, magnitude response and phase response</li> <li>◦ Analysis of LTI-systems using pole-zero plots</li> <li>◦ Allpass filters</li> <li>◦ Minimum-phase, maximum-phase and mixed phase filters</li> <li>◦ Stable systems</li> </ul> </li> <li>• Sampling <ul style="list-style-type: none"> <li>◦ Sampling theorem</li> <li>◦ Reconstruction of continuous-time signals in frequency domain and time domain</li> <li>◦ Oversampling</li> <li>◦ Aliasing</li> <li>◦ Sampling with pulses of finite duration, sample and hold</li> <li>◦ Decimation and interpolation</li> </ul> </li> <li>• Discrete-Time Fourier Transform (DTFT) <ul style="list-style-type: none"> <li>◦ Relation of Fourier transform and DTFT</li> <li>◦ Properties of the DTFT</li> </ul> </li> <li>• Discrete Fourier Transform (DFT) <ul style="list-style-type: none"> <li>◦ Relation of DTFT and DFT</li> <li>◦ Cyclic properties of the DFT</li> <li>◦ DFT matrix</li> <li>◦ Zero padding</li> <li>◦ Cyclic convolution</li> <li>◦ Fast Fourier Transform (FFT)</li> <li>◦ Application of the DFT: Orthogonal Frequency Division Multiplex (OFDM)</li> </ul> </li> <li>• Z-Transform <ul style="list-style-type: none"> <li>◦ Relation of Laplace transform, DTFT, and z-transform</li> <li>◦ Properties of the z-transform</li> <li>◦ Z-transform of some basic discrete-time signals</li> </ul> </li> <li>• Discrete-time systems, digital filters <ul style="list-style-type: none"> <li>◦ FIR and IIR filters</li> <li>◦ Z-transform of digital filters</li> <li>◦ Analysis of discrete-time systems using pole-zero plots in the z-domain</li> <li>◦ Stability</li> <li>◦ Allpass filters</li> <li>◦ Minimum-phase, maximum-phase and mixed-phase filters</li> <li>◦ Linear phase filters</li> </ul> </li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• T. Frey , M. Bossert , Signal- und Systemtheorie, B.G. Teubner Verlag 2004</li> <li>• K. Kammeyer, K. Kroschel, Digitale Signalverarbeitung, Teubner Verlag.</li> <li>• B. Girod ,R. Rabensteiner , A. Stenger , Einführung in die Systemtheorie, B.G. Teubner, Stuttgart, 1997</li> <li>• J.R. Ohm, H.D. Lüke , Signalübertragung, Springer-Verlag 8. Auflage, 2002</li> <li>• S. Haykin, B. van Veen: Signals and systems. Wiley.</li> <li>• Oppenheim, A.S. Willsky: Signals and Systems. Pearson.</li> </ul>

- Oppenheim, R. W. Schafer: Discrete-time signal processing. Pearson.

Course L0433: Signals and Systems	
<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>	Prof. Gerhard Bauch
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0803: Embedded Systems				
Courses				
Title	Typ		Hrs/wk	CP
Embedded Systems (L0805)	Lecture		3	3
Embedded Systems (L2938)	Project/problem-based Learning		1	1
Embedded Systems (L0806)	Recitation Section (small)		1	2
<b>Module Responsible</b>	Prof. Heiko Falk			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Computer Engineering			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	<p><i>Knowledge</i> Embedded systems can be defined as information processing systems embedded into enclosing products. This course teaches the foundations of such systems. In particular, it deals with an introduction into these systems (notions, common characteristics) and their specification languages (models of computation, hierarchical automata, specification of distributed systems, task graphs, specification of real-time applications, translations between different models).</p> <p>Another part covers the hardware of embedded systems: Sensors, A/D and D/A converters, real-time capable communication hardware, embedded processors, memories, energy dissipation, reconfigurable logic and actuators. The course also features an introduction into real-time operating systems, middleware and real-time scheduling. Finally, the implementation of embedded systems using hardware/software co-design (hardware/software partitioning, high-level transformations of specifications, energy-efficient realizations, compilers for embedded processors) is covered.</p> <p><i>Skills</i> After having attended the course, students shall be able to realize simple embedded systems. The students shall realize which relevant parts of technological competences to use in order to obtain a functional embedded systems. In particular, they shall be able to compare different models of computations and feasible techniques for system-level design. They shall be able to judge in which areas of embedded system design specific risks exist.</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 and to associate this knowledge with other classes.</p>			
<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 %	Subject	theoretical and practical work
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 minutes, contents of course and labs			
<b>Assignment for the Following Curricula</b>	<p>General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory</p> <p>Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory</p> <p>Electrical Engineering: Core Qualification: Elective Compulsory</p> <p>Engineering Science: Specialisation Mechatronics: Elective Compulsory</p> <p>Engineering Science: Specialisation Electrical Engineering: Elective Compulsory</p> <p>Aircraft Systems Engineering: Core Qualification: Elective Compulsory</p> <p>General Engineering Science (English program, 7 semester): Specialisation Mechatronics: Elective Compulsory</p> <p>Computer Science in Engineering: Core Qualification: Compulsory</p> <p>Mechatronics: Specialisation System Design: Elective Compulsory</p> <p>Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory</p> <p>Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory</p>			

Course L0805: Embedded Systems	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 48, Study Time in Lecture 42
<b>Lecturer</b>	Prof. Heiko Falk
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Specifications and Modeling</li> <li>• Embedded/Cyber-Physical Systems Hardware</li> <li>• System Software</li> <li>• Evaluation and Validation</li> <li>• Mapping of Applications to Execution Platforms</li> <li>• Optimization</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., Springer, 2012.</li> </ul>

Course L2938: Embedded Systems	
<b>Typ</b>	Project-/problem-based Learning
<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. Heiko Falk
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Specifications and Modeling</li> <li>• Embedded/Cyber-Physical Systems Hardware</li> <li>• System Software</li> <li>• Evaluation and Validation</li> <li>• Mapping of Applications to Execution Platforms</li> <li>• Optimization</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., Springer, 2012.</li> </ul>

Course L0806: Embedded Systems	
<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. Heiko Falk
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0727: Stochastics				
Courses				
Title			Typ	Hrs/wk CP
Stochastics (L0777)			Lecture	2 4
Stochastics (L0778)			Recitation Section (small)	2 2
<b>Module Responsible</b>	Prof. Matthias Schulte			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Calculus</li> <li>• Discrete algebraic structures (combinatorics)</li> <li>• Propositional logic</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 Stochastics. 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>Skills</i> <ul style="list-style-type: none"> <li>• Students can model problems from stochastics 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> <i>Social Competence</i> <ul style="list-style-type: none"> <li>• Students are able to work together (e.g. on their regular home work) in heterogeneously composed teams (i.e., teams from different study programs and background knowledge) 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> <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 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>	Written exam			
<b>Examination duration and scale</b>	120 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory General Engineering Science (German program, 7 semester): Specialisation Advanced Materials: Elective Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Specialisation Advanced Materials: Elective Compulsory Engineering Science: Specialisation Electrical Engineering: Elective Compulsory Computer Science in Engineering: Core Qualification: Compulsory Logistics and Mobility: Specialisation Engineering Science: Elective Compulsory Logistics and Mobility: Specialisation Information Technology: Elective Compulsory Orientation Studies: Core Qualification: Elective Compulsory Theoretical Mechanical Engineering: Core Qualification: Elective Compulsory Engineering and Management - Major in Logistics and Mobility: Specialisation Information Technology: Elective Compulsory			

Course L0777: Stochastics	
<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 Schulte
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Definitions of probability, conditional probability</li> <li>• Random variables</li> <li>• Independence</li> <li>• Distributions and density functions</li> <li>• Characteristics: expectation, variance, standard deviation, moments</li> <li>• Multivariate distributions</li> <li>• Law of large numbers and central limit theorem</li> <li>• Basic notions of stochastic processes</li> <li>• Basic concepts of statistics (point estimators, confidence intervals, hypothesis testing)</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• L. Dümbgen (2003): Stochastik für Informatiker, Springer.</li> <li>• H.-O. Georgii (2012): Stochastics: Introduction to Probability and Statistics, 2nd edition, De Gruyter.</li> <li>• N. Henze (2018): Stochastik für Einsteiger, 12th edition, Springer.</li> <li>• A. Klenke (2014): Probability Theory: A Comprehensive Course, 2nd edition, Springer.</li> <li>• U. Krengel (2005): Einführung in die Wahrscheinlichkeitstheorie und Statistik, 8th edition, Vieweg.</li> <li>• A.N. Shiryaev (2012): Problems in probability, Springer.</li> </ul>

Course L0778: Stochastics	
<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>	Prof. Matthias Schulte
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0833: Introduction to Control Systems			
Courses			
Title	Typ	Hrs/wk	CP
Introduction to Control Systems (L0654)	Lecture	2	4
Introduction to Control Systems (L0655)	Recitation Section (small)	2	2
<b>Module Responsible</b>	NN		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Representation of signals and systems in time and frequency domain, Laplace transform		
<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 represent dynamic system behavior in time and frequency domain, and can in particular explain properties of first and second order systems</li> <li>They can explain the dynamics of simple control loops and interpret dynamic properties in terms of frequency response and root locus</li> <li>They can explain the Nyquist stability criterion and the stability margins derived from it.</li> <li>They can explain the role of the phase margin in analysis and synthesis of control loops</li> <li>They can explain the way a PID controller affects a control loop in terms of its frequency response</li> <li>They can explain issues arising when controllers designed in continuous time domain are implemented digitally</li> </ul> <i>Skills</i> <ul style="list-style-type: none"> <li>Students can transform models of linear dynamic systems from time to frequency domain and vice versa</li> <li>They can simulate and assess the behavior of systems and control loops</li> <li>They can design PID controllers with the help of heuristic (Ziegler-Nichols) tuning rules</li> <li>They can analyze and synthesize simple control loops with the help of root locus and frequency response techniques</li> <li>They can calculate discrete-time approximations of controllers designed in continuous-time and use it for digital implementation</li> <li>They can use standard software tools (Matlab Control Toolbox, Simulink) for carrying out these tasks</li> </ul> <b>Personal Competence</b> <i>Social Competence</i> Students can work in small groups to jointly solve technical problems, and experimentally validate their controller designs <i>Autonomy</i> Students can obtain information from provided sources (lecture notes, software documentation, experiment guides) and use it when solving given problems. They can assess their knowledge in weekly on-line tests and thereby control their learning progress.			
<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>	General Engineering Science (German program, 7 semester): Core Qualification: Compulsory Bioprocess Engineering: Core Qualification: Compulsory Chemical and Bioprocess Engineering: Core Qualification: Compulsory Data Science: Core Qualification: Elective Compulsory Data Science: Specialisation II. Application: Elective Compulsory Electrical Engineering: Core Qualification: Compulsory Green Technologies: Energy, Water, Climate: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Integrated Building Technology: Core Qualification: Elective Compulsory Logistics and Mobility: Specialisation Information Technology: Elective Compulsory Logistics and Mobility: Specialisation Traffic Planning and Systems: Elective Compulsory Logistics and Mobility: Specialisation Production Management and Processes: Elective Compulsory Mechanical Engineering: Core Qualification: Compulsory Mechatronics: Core Qualification: Compulsory Technomathematics: Specialisation III. Engineering Science: Elective Compulsory Theoretical Mechanical Engineering: Technical Complementary Course Core Studies: Elective Compulsory Process Engineering: Core Qualification: Compulsory Engineering and Management - Major in Logistics and Mobility: Specialisation Information Technology: Elective Compulsory Engineering and Management - Major in Logistics and Mobility: Specialisation Traffic Planning and Systems: Elective Compulsory Engineering and Management - Major in Logistics and Mobility: Specialisation Production Management and Processes: Elective Compulsory		

Course L0654: Introduction to Control Systems	
<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>	NN
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<p>Signals and systems</p> <ul style="list-style-type: none"> <li>• Linear systems, differential equations and transfer functions</li> <li>• First and second order systems, poles and zeros, impulse and step response</li> <li>• Stability</li> </ul> <p>Feedback systems</p> <ul style="list-style-type: none"> <li>• Principle of feedback, open-loop versus closed-loop control</li> <li>• Reference tracking and disturbance rejection</li> <li>• Types of feedback, PID control</li> <li>• System type and steady-state error, error constants</li> <li>• Internal model principle</li> </ul> <p>Root locus techniques</p> <ul style="list-style-type: none"> <li>• Root locus plots</li> <li>• Root locus design of PID controllers</li> </ul> <p>Frequency response techniques</p> <ul style="list-style-type: none"> <li>• Bode diagram</li> <li>• Minimum and non-minimum phase systems</li> <li>• Nyquist plot, Nyquist stability criterion, phase and gain margin</li> <li>• Loop shaping, lead lag compensation</li> <li>• Frequency response interpretation of PID control</li> </ul> <p>Time delay systems</p> <ul style="list-style-type: none"> <li>• Root locus and frequency response of time delay systems</li> <li>• Smith predictor</li> </ul> <p>Digital control</p> <ul style="list-style-type: none"> <li>• Sampled-data systems, difference equations</li> <li>• Tustin approximation, digital implementation of PID controllers</li> </ul> <p>Software tools</p> <ul style="list-style-type: none"> <li>• Introduction to Matlab, Simulink, Control toolbox</li> <li>• Computer-based exercises throughout the course</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Werner, H., Lecture Notes „Introduction to Control Systems“</li> <li>• G.F. Franklin, J.D. Powell and A. Emami-Naeini "Feedback Control of Dynamic Systems", Addison Wesley, Reading, MA, 2009</li> <li>• K. Ogata "Modern Control Engineering", Fourth Edition, Prentice Hall, Upper Saddle River, NJ, 2010</li> <li>• R.C. Dorf and R.H. Bishop, "Modern Control Systems", Addison Wesley, Reading, MA 2010</li> </ul>

Course L0655: Introduction to Control Systems	
<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>	NN
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0675: Introduction to Communications and Random Processes				
Courses				
Title		Typ	Hrs/wk	CP
Introduction to Communications and Random Processes (L0442)		Lecture	3	4
Introduction to Communications and Random Processes (L0443)		Recitation Section (large)	1	1
Introduction to Communications and Random Processes (L2354)		Recitation Section (small)	1	1
Module Responsible	Prof. Gerhard Bauch			
Admission Requirements	None			
Recommended Previous Knowledge	<ul style="list-style-type: none"><li>Mathematics 1-3</li><li>Signals and Systems</li></ul>			
Educational Objectives	After taking part successfully, students have reached the following learning results			
Professional Competence	<div><div>Knowledge</div><div>The students know and understand the fundamental building blocks of a communications system. They can describe and analyse the individual building blocks using knowledge of signal and system theory as well as the theory of stochastic processes. The are aware of the essential resources and evaluation criteria of information transmission and are able to design and evaluate a basic communications system.</div><div>The students are familiar with the contents of lecture and tutorials. They can explain and apply them to new problems.</div><div>Skills</div><div>The students are able to design and evaluate a basic communications system. In particular, they can estimate the required resources in terms of bandwidth and power. They are able to assess essential evaluation parameters of a basic communications system such as bandwidth efficiency or bit error rate and to decide for a suitable transmission method.</div><div>Personal Competence</div><div>Social Competence</div><div>The students can jointly solve specific problems.</div><div>Autonomy</div><div>The students are able to acquire relevant information from appropriate literature sources. They can control their level of knowledge during the lecture period by solving tutorial problems, software tools, clicker system.</div></div>			
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70			
Credit points	6			
Course achievement	None			
Examination	Written exam			
Examination duration and scale	90 min			
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Electrical Engineering: Compulsory Data Science: Core Qualification: Elective Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Electrical Engineering: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Mechatronics: Specialisation Electrical Systems: Compulsory Technomathematics: Specialisation III. Engineering Science: Elective Compulsory			

Course L0442: Introduction to Communications and Random Processes	
<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. Gerhard Bauch
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>Introduction to communications engineering</li> <li>Open Systems Interconnection (OSI) reference model</li> <li>Components of a digital communications system</li> <li>Fundamentals of signals and systems               <ul style="list-style-type: none"> <li>Analog and digital signals</li> <li>Principles of Analog-to-digital (A/D) conversion</li> <li>Deterministic and random signals</li> <li>Power and energy of signals</li> <li>Linear time-invariant (LTI) systems</li> <li>Quadrature amplitude modulation (QAM)</li> </ul> </li> <li>Introduction to stochastics</li> <li>Probability theory               <ul style="list-style-type: none"> <li>Random experiments</li> <li>Probability model, probability space, sample space</li> <li>Definitions of probability                   <ul style="list-style-type: none"> <li>Probability according to Bernoulli/Laplace</li> <li>Probability according to van Mises, relative frequency</li> <li>Bertrand's paradox</li> <li>Axiomatic definition of probability according to Kolmogorov</li> <li>Probability of disjoint and non-disjoint events</li> <li>Venn diagrams</li> </ul> </li> </ul> </li> </ul>

- Continuous and discrete random variables
  - Probability density function (pdf), cumulative distribution function (cdf)
  - Expected value, mean, median, quadratic mean, variance, standard deviation, higher moments
  - Examples for probability distributions (Bernoulli distribution, two-point distribution, uniform distribution, Gaussian (normal) distribution, Rayleigh distribution, etc.)
- Multiple random variables
  - Conditional probability, joint probability
  - Conditional and joint probability density function
  - Bayes' rule
  - Correlation coefficient
  - Two-dimensional Gaussian distribution
  - Statistically independent, uncorrelated and orthogonal random variables
  - Independent identically distributed (iid) random variables
  - Properties of expected value and variance
  - Covariance
  - Probability density function (pdf) and cumulative distribution function (cdf) of the sum of statistically independent random variables
  - Central limit theorem
- Probability density functions (pdfs) in data transmission
- Continuous-time and discrete-time random processes
  - Examples for random processes
  - Ensemble average and time average
  - Ergodic random processes
  - Quadratic mean and variance
  - Probability density function (pdf) and cumulative distribution function (cdf)
  - Joint probability density function (pdf) and joint cumulative distribution function (cdf)
  - Statistically independent, uncorrelated and orthogonal random processes
  - Stationary random processes
  - Correlation functions: Autocorrelation function, crosscorrelation function, average autocorrelation function of non-stationary random processes, autocorrelation and crosscorrelation function of stationary processes, autocovariance function, crosscovariance function
  - Autocorrelation matrix, crosscorrelation matrix, autocovariance matrix, crosscovariance matrix
  - Pseudo-noise sequences, example: Code division multiple access (CDMA)
  - Autocorrelation function, power spectral density (psd), signal power, Einstein-Wiener-Khinchine relations
  - White (Gaussian) noise
- Filtering of random processes by LTI systems
  - Transformation of the probability density function (pdf)
  - Transformation of the mean
  - Transformation of the power spectral density (psd)
  - Correlation functions of input and output signal
  - Filtering of white Gaussian noise
  - Bandlimitation for noise power limitation
  - Preemphasis and deemphasis
- Companding, mu-law, A-law
- Functions of random variables
  - Transformation of probabilities and of the probability density function (pdf)
  - Application: Non-linear amplifiers
- Functions of two random variables
  - Probability density function
  - Examples: Rayleigh distribution, magnitude of an OFDM signal, magnitude of a received radio signal
- Transmission channels and channel models
  - Wireline channels: Telephone cable, coaxial cable, optical fiber
  - Wireless channels: Fading radio channel, underwater channels
  - Frequency-flat and frequency-selective channels
  - Additive white Gaussian noise (AWGN) channel
  - Signal to noise power ratio (SNR)
  - Discrete-time channel models
  - Discrete memoryless channels (DMC)
- Analog-to-digital conversion
  - Sampling
    - Sampling theorem
  - Pulse modulation
    - Pulse-amplitude modulation (PAM)
    - Pulse-duration modulation (PDM), pulse-width modulation (PWM)
    - Pulse-position modulation (PPM)
    - Pulse-code modulation (PCM)
  - Quantization
    - Linear quantization, midtread and midrise characteristic
    - Quantization error, quantization noise
    - Signal-to-quantization noise ratio
    - Non-linear quantization, compressor characteristics, mu-law, A-law
    - Speech transmission with PCM
  - Differential pulse-code modulation (DPCM)
    - Linear prediction according to the minimum mean squared error (MMSE) criterion.
    - DPCM with forward prediction and backward prediction

	<ul style="list-style-type: none"> <li>▪ SNR gain of DPCM over PCM</li> <li>▪ Delta modulation</li> <li>• Fundamentals of information theory and coding <ul style="list-style-type: none"> <li>◦ Definitions of information: Self-information, entropy</li> <li>◦ Binary entropy function</li> <li>◦ Source coding theorem</li> <li>◦ Source coding: Huffman code</li> <li>◦ Mutual information and channel capacity</li> <li>◦ Channel capacity of the AWGN channel and the binary input AWGN channel</li> <li>◦ Channel coding theorem</li> <li>◦ Principles of channel coding: Code rate and data rate, Hamming distance, minimum Hamming distance, error detection and error correction</li> <li>◦ Examples for channel codes: Block codes and convolutional codes, repetition code, single parity check code, Hamming code, Turbo codes</li> </ul> </li> <li>• Combinatorics <ul style="list-style-type: none"> <li>◦ Variation with and without repetition</li> <li>◦ Combination with and without repetition</li> <li>◦ Permutation, Permutation of multisets</li> <li>◦ Word error probabilities of linear block codes</li> </ul> </li> <li>• Baseband transmission <ul style="list-style-type: none"> <li>◦ Pulse shaping: Non-return to zero (NRZ) rectangular pulses, Manchester pulses, raised-cosine pulses, square-root raised-cosine pulses, Gaussian pulses</li> <li>◦ Transmit signal energy, average energy per symbol</li> <li>◦ Power spectral density (psd) of baseband signals</li> <li>◦ Definitions of signal bandwidth</li> <li>◦ Bandwidth efficiency</li> <li>◦ Intersymbol interference (ISI)</li> <li>◦ First and second Nyquist criterion</li> <li>◦ Eye patterns</li> <li>◦ Receive filter design: Matched filter</li> <li>◦ Matched-filter receiver and correlation receiver</li> <li>◦ Square-root Nyquist pulse shaping</li> <li>◦ Discrete-time AWGN channel model</li> </ul> </li> <li>• Maximum a posteriori probability (MAP) and maximum likelihood (ML) detection</li> <li>• Bit error probability in AWGN channels for binary antipodal and on-off signaling</li> <li>• Band-pass transmission via carrier modulation <ul style="list-style-type: none"> <li>◦ Amplitude modulation, frequency modulation, phase modulation</li> <li>◦ Linear digital modulation methods: On-off keying (OOK), phase-shift keying (PSK), amplitude shift keying (ASK), quadrature amplitude shift keying (QAM)</li> </ul> </li> <li>•</li> </ul>
<b>Literature</b>	<p>K. Kammeyer: Nachrichtenübertragung, Teubner</p> <p>P.A. Höher: Grundlagen der digitalen Informationsübertragung, Teubner.</p> <p>M. Bossert: Einführung in die Nachrichtentechnik, Oldenbourg.</p> <p>J.G. Proakis, M. Salehi: Grundlagen der Kommunikationstechnik. Pearson Studium.</p> <p>J.G. Proakis, M. Salehi: Digital Communications. McGraw-Hill.</p> <p>S. Haykin: Communication Systems. Wiley</p> <p>J.G. Proakis, M. Salehi: Communication Systems Engineering. Prentice-Hall.</p> <p>J.G. Proakis, M. Salehi, G. Bauch, Contemporary Communication Systems. Cengage Learning.</p>

Course L0443: Introduction to Communications and Random Processes	
<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. Gerhard Bauch
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L2354: Introduction to Communications and Random Processes	
<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. Gerhard Bauch
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1431: Practical Course IIW				
Courses				
Title	Typ		Hrs/wk	CP
Practical Course IIW (L2160)	Project-/problem-based Learning		8	6
Module Responsible	Prof. Görschwin Fey			
Admission Requirements	None			
Recommended Previous Knowledge	<p>Successful participation in the modules:</p> <ul style="list-style-type: none"> <li>• Procedural Programming</li> <li>• Algorithms and Data Structures</li> <li>• Embedded Systems</li> <li>• Computer Engineering</li> <li>• Electrical Engineering I</li> <li>• Signals and Systems</li> </ul>			
Educational Objectives	After taking part successfully, students have reached the following learning results			
Professional Competence	<p><i>Knowledge</i> Students get to know tools used by development teams to</p> <ul style="list-style-type: none"> <li>• application-driven software development</li> <li>• deriving requirements and models according to engineering disciplines</li> <li>• software plan development flows,</li> <li>• manage task distribution,</li> <li>• manage source code, and</li> <li>• test software.</li> </ul> <p><i>Skills</i> Students work in teams on a larger project. The required competences are learned and practically applied. These are for example:</p> <ul style="list-style-type: none"> <li>• specifying software based on user requirements</li> <li>• implementing the interaction of a computer system with the physical environment</li> <li>• creating a software architecture</li> <li>• implementing and testing software in a team, and</li> <li>• using the related development tools.</li> </ul> <p><i>Personal Competence</i></p> <p><i>Social Competence</i> Team work has its own challenges with respect to interaction of team members as well as finding the necessary agreement during joint software development. During the project students learn the required competences and experience the practical needs.</p> <p><i>Autonomy</i> During team work it is mandatory to take and explain a certain position, to independently complete assigned tasks, and to present results to the team. Open issues must be identified and returned into the team to find an agreed resolution.</p>			
Workload in Hours				
Credit points				
Course achievement				
Examination	Subject theoretical and practical work			
Examination duration and scale	Evaluation of engagement, project report and final presentation			
Assignment for the Following Curricula	Computer Science in Engineering: Core Qualification: Compulsory			

Course L2160: Practical Course IIW	
Typ	Project-/problem-based Learning
Hrs/wk	8
CP	6
Workload in Hours	Independent Study Time 68, Study Time in Lecture 112
Lecturer	NN, Dozenten des SD E
Language	DE/EN
Cycle	WiSe
Content	<p>Bridging the gap between disciplines and moving from theory to practice are essential in the Computer Science in Engineering programme. Exactly the relevant skills are learned in the IIW internship. A software program, an embedded system or cyber physical system is developed during the course of the project. The respective lecturer provides the concrete task description. Participating students work as a team to solve the task. This induces a typical project flow as it occurs in enterprises as well. Typical steps like defining a specification, creating a hardware-software-architecture as well as implementation and testing are mandatory. Students are also responsible for project planning, defining and assigning sub tasks to team members. Common development tools supporting planning, management and realization are used within the project.</p> <p>The project is split into regular plenary sessions and into independent team work.</p>
Literature	<p>Wird durch die jeweiligen DozentInnen zur Verfügung gestellt.</p> <p>Supplied by the respective lecturer.</p>

## Specialization I. Computer Science

### Module M0731: Functional Programming

#### Courses

Title	Type	Hrs/wk	CP
Functional Programming (L0624)	Lecture	2	2
Functional Programming (L0625)	Recitation Section (large)	2	2
Functional Programming (L0626)	Recitation Section (small)	2	2

<b>Module Responsible</b>	Prof. Sibylle Schupp		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Discrete mathematics at high-school level		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	Students apply the principles, constructs, and simple design techniques of functional programming. They demonstrate their ability to read Haskell programs and to explain Haskell syntax as well as Haskell's read-eval-print loop. They interpret warnings and find errors in programs. They apply the fundamental data structures, data types, and type constructors. They employ strategies for unit tests of functions and simple proof techniques for partial and total correctness. They distinguish laziness from other evaluation strategies.		
<i>Skills</i>	Students break a natural-language description down in parts amenable to a formal specification and develop a functional program in a structured way. They assess different language constructs, make conscious selections both at specification and implementations level, and justify their choice. They analyze given programs and rewrite them in a controlled way. They design and implement unit tests and can assess the quality of their tests. They argue for the correctness of their program.		
<b>Personal Competence</b>			
<i>Social Competence</i>	Students practice peer programming with varying peers. They explain problems and solutions to their peer. They defend their programs orally. They communicate in English.		
<i>Autonomy</i>	In programming labs, students learn under supervision (a.k.a. "Betreutes Programmieren") the mechanics of programming. In exercises, they develop solutions individually and independently, and receive feedback.		
<b>Workload in Hours</b>	Independent Study Time 96, Study Time in Lecture 84		
<b>Credit points</b>	6		
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>
	Yes	15 %	Exercises
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 min		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Elective Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Engineering Science: Specialisation Mechatronics: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechatronics: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory		

Course L0624: Functional Programming	
<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. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions</li> <li>• Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions</li> <li>• Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type</li> <li>• Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps)</li> <li>• Modules</li> <li>• Interactive Programming</li> <li>• Lazy Evaluation, Call-by-Value, Strictness</li> <li>• Design Recipes</li> <li>• Testing (axiom-based, invariant-based, against reference implementation)</li> <li>• Reasoning about Programs (equation-based, inductive)</li> <li>• Idioms of Functional Programming</li> <li>• Haskell Syntax and Semantics</li> </ul>
<b>Literature</b>	Graham Hutton, Programming in Haskell, Cambridge University Press 2007.

Course L0625: Functional Programming	
<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. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions</li> <li>• Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions</li> <li>• Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type</li> <li>• Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps)</li> <li>• Modules</li> <li>• Interactive Programming</li> <li>• Lazy Evaluation, Call-by-Value, Strictness</li> <li>• Design Recipes</li> <li>• Testing (axiom-based, invariant-based, against reference implementation)</li> <li>• Reasoning about Programs (equation-based, inductive)</li> <li>• Idioms of Functional Programming</li> <li>• Haskell Syntax and Semantics</li> </ul>
<b>Literature</b>	Graham Hutton, Programming in Haskell, Cambridge University Press 2007.

Course L0626: Functional Programming	
<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>	Prof. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions</li> <li>• Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions</li> <li>• Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type</li> <li>• Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps)</li> <li>• Modules</li> <li>• Interactive Programming</li> <li>• Lazy Evaluation, Call-by-Value, Strictness</li> <li>• Design Recipes</li> <li>• Testing (axiom-based, invariant-based, against reference implementation)</li> <li>• Reasoning about Programs (equation-based, inductive)</li> <li>• Idioms of Functional Programming</li> <li>• Haskell Syntax and Semantics</li> </ul>
<b>Literature</b>	Graham Hutton, Programming in Haskell, Cambridge University Press 2007.

Module M0625: Databases				
Courses				
Title		Typ	Hrs/wk	CP
Databases (L0337)		Lecture	3	4
Databases - Exercise (L1150)		Recitation Section (small)	2	2
Module Responsible	Prof. Stefan Schulte			
Admission Requirements	None			
Recommended Previous Knowledge	Students should have basic knowledge in the following areas: <ul style="list-style-type: none"><li>• Discrete Algebraic Structures</li><li>• Procedural Programming</li><li>• Automata Theory and Formal Languages</li><li>• Programming Paradigms</li></ul>			
Educational Objectives	After taking part successfully, students have reached the following learning results			
Professional Competence	<div>Knowledge</div> After successful completion of the course, students know: <ul style="list-style-type: none"><li>• Introduction to database systems</li><li>• Design instruments for relational databases, especially entity-relationship</li><li>• The relational model</li><li>• Relational query languages, especially SQL</li><li>• Normalization</li><li>• Physical data organization</li><li>• Transaction management</li><li>• Query optimization</li><li>• Data representation</li><li>• Object-oriented and object-relational databases</li><li>• Paradigms and concepts of current technologies for data modelling and database systems</li></ul> <div>Skills</div> The students acquire the ability to model a database and to work with it. This comprises especially the application of design methodologies and query and definition languages. Furthermore, students are able to apply basic functionalities needed to run a database. <div>Personal Competence</div> <div>Social Competence</div> 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. <div>Autonomy</div> Students are able to independently investigate a complex problem and assess which competencies are required to solve it.			
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70			
Credit points	6			
Course achievement	None			
Examination	Written exam			
Examination duration and scale	90 min			
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Data Science: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Specialisation Data Science: Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory			

Course L0337: Databases	
<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. Stefan Schulte
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction to database systems</li> <li>• Design instruments for relational databases, especially entity-relationship</li> <li>• The relational model</li> <li>• Relational query languages, especially SQL</li> <li>• Normalization</li> <li>• Physical data organization</li> <li>• Transaction management</li> <li>• Query optimization</li> <li>• Data representation</li> <li>• Object-oriented and object-relational databases</li> <li>• Paradigms and concepts of current technologies for data modelling and database systems</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• A. Kemper, A. Eickler, Datenbanksysteme, 10. Auflage, De Gruyter, Oldenbourg, 2015</li> <li>• R. Elmasri, S. B. Navathe, Fundamentals of Database Systems, 7th edition, Pearson, 2016</li> </ul>

Course L1150: Databases - Exercise	
<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>	Prof. Stefan Schulte
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction to database systems</li> <li>• Design instruments for relational databases, especially entity-relationship</li> <li>• The relational model</li> <li>• Relational query languages, especially SQL</li> <li>• Normalization</li> <li>• Physical data organization</li> <li>• Transaction management</li> <li>• Query optimization</li> <li>• Data representation</li> <li>• Object-oriented and object-relational databases</li> <li>• Paradigms and concepts of current technologies for data modelling and database systems</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• A. Kemper, A. Eickler, Datenbanksysteme, 10. Auflage, De Gruyter, Oldenbourg, 2015</li> <li>• R. Elmasri, S. B. Navathe, Fundamentals of Database Systems, 7th edition, Pearson, 2016</li> </ul>

Module M0791: Computer Architecture				
Courses				
Title	Typ		Hrs/wk	CP
Computer Architecture (L0793)	Lecture		2	3
Computer Architecture (L0794)	Project-/problem-based Learning		2	2
Computer Architecture (L1864)	Recitation Section (small)		1	1
<b>Module Responsible</b>	Prof. Heiko Falk			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Module "Computer Engineering"			
<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 presents advanced concepts from the discipline of computer architecture. In the beginning, a broad overview over various programming models is given, both for general-purpose computers and for special-purpose machines (e.g., signal processors). Next, foundational aspects of the micro-architecture of processors are covered. Here, the focus particularly lies on the so-called pipelining and the methods used for the acceleration of instruction execution used in this context. The students get to know concepts for dynamic scheduling, branch prediction, superscalar execution of machine instructions and for memory hierarchies.</p> <p><i>Skills</i> The students are able to describe the organization of processors. They know the different architectural principles and programming models. The students examine various structures of pipelined processor architectures and are able to explain their concepts and to analyze them w.r.t. criteria like, e.g., performance or energy efficiency. They evaluate different structures of memory hierarchies, know parallel computer architectures and are able to distinguish between instruction- and data-level parallelism.</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 and to associate this knowledge with other classes.</p>			
<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	15 %	Subject	theoretical and practical work
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 minutes, contents of course and 4 attestations from the PBL "Computer architecture"			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Aircraft Systems Engineering: Core Qualification: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Aeronautics: Core Qualification: Elective Compulsory Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory			

Course L0793: Computer Architecture	
<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. Heiko Falk
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• VHDL Basics</li> <li>• Programming Models</li> <li>• Realization of Elementary Data Types</li> <li>• Dynamic Scheduling</li> <li>• Branch Prediction</li> <li>• Superscalar Machines</li> <li>• Memory Hierarchies</li> </ul> <p>The theoretical tutorials amplify the lecture's content by solving and discussing exercise sheets and thus serve as exam preparation. Practical aspects of computer architecture are taught in the FPGA-based PBL on computer architecture whose attendance is mandatory.</p>
<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> </ul>

Course L0794: Computer Architecture	
<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. Heiko Falk
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L1864: Computer Architecture	
<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. Heiko Falk
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1883: Introduction to Quantum Computing				
Courses				
Title	Typ		Hrs/wk	CP
Introduction to Quantum Computing (L3109)	Lecture		2	3
Introduction to Quantum Computing (L3110)	Recitation Section (large)		2	3
<b>Module Responsible</b>	Prof. Martin Kliesch			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Linear algebra and very good mathematical skills</li> <li>Prior knowledge in theoretical computer science or quantum mechanics is helpful but not required</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>Information theoretic understanding of quantum mechanics</li> <li>The quantum teleportation protocol</li> <li>Basic quantum algorithms</li> <li>Grover's search algorithm</li> <li>The quantum Fourier transform and Shor's algorithm for integer factoring</li> <li>The unitary circuit model of quantum computation (qubits, quantum gates and readout) and the complexity class BQP</li> </ul>			
<i>Knowledge</i>				
<i>Skills</i>				
<b>Personal Competence</b>				
<i>Social Competence</i>	After completing this module, students are expected to be able to work on subject-specific tasks alone or in a group and to present the results appropriately. Moreover, students will be trained to identify and defuse misleading statements related to quantum computing, which can often be found in popular media.			
<i>Autonomy</i>	After completion of this module, students are able to work out sub-areas of the subject independently using textbooks and other literature, to summarize and present the acquired knowledge and to link it to the contents of other courses.			
<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	20 %	Exercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory			

Course L3109: Introduction to Quantum Computing	
<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. Martin Kliesch
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<p>Quantum computing is among the most exciting applications of quantum mechanics. Quantum algorithms can solve computational problems efficiently that have a prohibitive runtime on traditional computers. Such problems include, for instance, factoring of integer numbers or energy estimation problems from quantum chemistry and material science.</p> <p>This course provides an introduction to the topic. An emphasize will be put on conceptual and mathematical aspects.</p>
<b>Literature</b>	<ul style="list-style-type: none"> <li>Course specific lecture notes will be provided</li> <li>Nielsen and Chuang, Quantum Computation and Quantum Information</li> <li>Sevag Gharibian's lecture notes</li> </ul>

Course L3110: Introduction to Quantum Computing	
<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. Martin Kliesch
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0562: Computability and Complexity Theory				
Courses				
Title	Typ		Hrs/wk	CP
Computability and Complexity Theory (L0166)	Lecture		2	3
Computability and Complexity Theory (L0167)	Recitation Section (small)		2	3
<b>Module Responsible</b>	Prof. Martin Kliesch			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Discrete Algebraic Structures, Automata Theory, Logic, and Formal Language Theory			
<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>• Basic models of computation (finite state machines, Turing machines)</li> <li>• Decision problems and formal languages</li> <li>• Gödel numbering of computations</li> <li>• Universal computability</li> <li>• Decidable and undecidable problems</li> <li>• Reductions, diagonalization, Rice's theorem</li> <li>• Time and space complexity</li> <li>• The complexity classes P and NP</li> <li>• Hierarchy theorems</li> <li>• Polynomial time reductions, NP-completeness</li> <li>• Cook-Levin theorem</li> <li>• Uniform circuit families</li> </ul>			
<i>Skills</i>	After completing this module, students are able to <ul style="list-style-type: none"> <li>• reproduce the knowledge taught in the course,</li> <li>• reproduce simpler proofs of the course and reproduce the ideas of the more complicated ones,</li> <li>• establish connections between the concepts taught, and</li> <li>• apply the learned knowledge to concrete problems.</li> </ul>			
<b>Personal Competence</b> <i>Social Competence</i>	After completing this module, students are able to work on subject-specific tasks alone or in a group and to present the results appropriately.			
<i>Autonomy</i>	After completion of this module, students are able to work out sub-areas of the subject area independently on the basis of textbooks and other literature, to summarize and present the acquired knowledge and to link it to the contents of other courses.			
<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	15 %	Exercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory General Engineering Science (German program, 7 semester): Specialisation Data Science: Elective Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Elective Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory			

Course L0166: Computability and Complexity Theory	
<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. Martin Kliesch
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	
<b>Literature</b>	

Course L0167: Computability and Complexity Theory	
<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. Martin Kliesch
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0754: Compiler Construction				
Courses				
Title			Typ	Hrs/wk CP
Compiler Construction (L0703)			Lecture	2 2
Compiler Construction (L0704)			Recitation Section (small)	2 4
<b>Module Responsible</b>	Prof. Sibylle Schupp			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Practical programming experience</li> <li>• Automata theory and formal languages</li> <li>• Functional programming or procedural programming</li> <li>• Object-oriented programming, algorithms, and data structures</li> <li>• Basic knowledge of software engineering</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 explain the workings of a compiler and break down a compilation task in different phases. They apply and modify the major algorithms for compiler construction and code improvement. They can re-write those algorithms in a programming language, run and test them. They choose appropriate internal languages and representations and justify their choice. They explain and modify implementations of existing compiler frameworks and experiment with frameworks and tools.</p> <p><i>Skills</i> Students design and implement arbitrary compilation phases. They integrate their code in existing compiler frameworks. They organize their compiler code properly as a software project. They generalize algorithms for compiler construction to algorithms that analyze or synthesize software.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> Students develop the software in a team. They explain problems and solutions to their team members. They present and defend their software in class. They communicate in English.</p> <p><i>Autonomy</i> Students develop their software independently and define milestones by themselves. They receive feedback throughout the entire project. They organize the software project so that they can assess their progress themselves.</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 (Compiler)			
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory			

Course L0703: Compiler Construction	
<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. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Lexical and syntactic analysis</li> <li>• Semantic analysis</li> <li>• High-level optimization</li> <li>• Intermediate languages and code generation</li> <li>• Compilation pipeline</li> </ul>
<b>Literature</b>	Alfred Aho, Jeffrey Ullman, Ravi Sethi, and Monica S. Lam, Compilers: Principles, Techniques, and Tools, 2nd edition  Aarne Ranta, Implementing Programming Languages, An Introduction to Compilers and Interpreters, with an appendix coauthored by Markus Forsberg, College Publications, London, 2012

Course L0704: Compiler Construction	
<b>Typ</b>	Recitation Section (small)
<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. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0732: Software Engineering				
Courses				
Title	Typ		Hrs/wk	CP
Software Engineering (L0627)	Lecture		2	3
Software Engineering (L0628)	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>Procedural programming or Functional programming</li> <li>Object-oriented programming, 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>	<p><i>Knowledge</i> Students explain the phases of the software life cycle, describe the fundamental terminology and concepts of software engineering, and paraphrase the principles of structured software development. They give examples of software-engineering tasks of existing large-scale systems. They write test cases for different test strategies and devise specifications or models using different notations, and critique both. They explain simple design patterns and the major activities in requirements analysis, maintenance, and project planning.</p> <p><i>Skills</i> For a given task in the software life cycle, students identify the corresponding phase and select an appropriate method. They choose the proper approach for quality assurance. They design tests for realistic systems, assess the quality of the tests, and find errors at different levels. They apply and modify non-executable artifacts. They integrate components based on interface specifications.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> Students practice peer programming. They explain problems and solutions to their peer. They communicate in English.</p> <p><i>Autonomy</i> Using on-line quizzes and accompanying material for self study, students can assess their level of knowledge continuously and adjust it appropriately. Working on exercise problems, they receive additional feedback.</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	15 %	Exercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Core Qualification: Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory			

Course L0627: 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. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>Model-based software engineering               <ul style="list-style-type: none"> <li>Information modeling (use case diagrams)</li> <li>Behavioral modeling (finite state machines, Petri Nets, behavioral UML diagrams)</li> <li>Structural modeling (OOA, UML class diagrams, OCL)</li> <li>Model-based testing</li> </ul> </li> <li>Engineering software products               <ul style="list-style-type: none"> <li>Agile processes</li> <li>Architecture</li> <li>Code-based testing</li> <li>System-level testing</li> </ul> </li> <li>Software management               <ul style="list-style-type: none"> <li>Maintenance</li> <li>Project management</li> <li>Software processes</li> </ul> </li> </ul>
<b>Literature</b>	Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, Pearson 2020.  Kassem A. Saleh, Software Engineering, J. Ross Publishing 2009.

Course L0628: Software Engineering	
<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>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1300: Software Development			
Courses			
Title	Typ	Hrs/wk	CP
Software Development (L1790)	Project-/problem-based Learning	2	5
Software Development (L1789)	Lecture	1	1
<b>Module Responsible</b>	Prof. Sibylle Schupp		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Introduction to Software Engineering</li> <li>• Programming Skills</li> <li>• Experience with Developing Small to Medium-Size Programs</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 fundamental concepts of agile methods, describe the process of test-driven development, and explain how continuous integration can be used in different scenarios. They give examples of selected pitfalls in software development, regarding scalability and other non-functional requirements. They write unit tests and build scripts and combine them in a corresponding integration environment. They explain major activities in requirements analysis, program comprehension, and agile project development.</p> <p><i>Skills</i></p> <p>For a given task on a legacy system, students identify the corresponding parts in the system and select an appropriate method for understanding the details. They choose the proper approach of splitting a task in independent testable and extensible pieces and, thus, solve the task with proper methods for quality assurance. They design tests for legacy systems, create automated builds, and find errors at different levels. They integrate the resulting artifacts in a continuous development environment</p>		
<b>Personal Competence</b>			
<i>Social Competence</i>			
<i>Autonomy</i>	<p>Students discuss different design decisions in a group. They defend their solutions orally. They communicate in English.</p> <p>Using accompanying tools, students can assess their level of knowledge continuously and adjust it appropriately. Within limits, they can set their own goals. Upon successful completion, students can identify and formulate concrete problems of software systems and propose solutions. Within this field, they can conduct independent studies to acquire the necessary competencies. They can devise plans to arrive at new solutions or assess existing ones.</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>	Software		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory		

Course L1790: Software Development	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	2
<b>CP</b>	5
<b>Workload in Hours</b>	Independent Study Time 122, 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>• Agile Methods</li> <li>• Test-Driven Development and Unit Testing</li> <li>• Continuous Integration</li> <li>• Web Services</li> <li>• Scalability</li> <li>• From Defects to Failure</li> </ul>
<b>Literature</b>	<p>Duvall, Paul M. Continuous Integration. Pearson Education India, 2007.</p> <p>Humble, Jez, and David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010.</p> <p>Martin, Robert Cecil. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003.</p> <p><a href="http://scrum-kompakt.de/">http://scrum-kompakt.de/</a></p> <p>Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley &amp; Sons, 2011.</p>

Course L1789: Software Development	
<b>Typ</b>	Lecture
<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. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Agile Methods</li> <li>• Test-Driven Development and Unit Testing</li> <li>• Continuous Integration</li> <li>• Web Services</li> <li>• Scalability</li> <li>• From Defects to Failure</li> </ul>
<b>Literature</b>	<p>Duvall, Paul M. Continuous Integration. Pearson Education India, 2007.</p> <p>Humble, Jez, and David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010.</p> <p>Martin, Robert Cecil. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003.</p> <p><a href="http://scrum-kompakt.de/">http://scrum-kompakt.de/</a></p> <p>Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley &amp; Sons, 2011.</p>

Module M1595: Machine Learning I				
Courses				
Title	Typ		Hrs/wk	CP
Machine Learning I (L2432)	Lecture		2	3
Machine Learning I (L2433)	Recitation Section (small)		3	3
<b>Module Responsible</b>	Prof. Nihat Ay			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Linear Algebra, Analysis, Basic Programming Course			
<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 know</p> <ul style="list-style-type: none"> <li>• general principles of machine learning learning: supervised/unsupervised learning, generative/descriptive learning, parametric/non-parametric learning</li> <li>• different learning methods: neural networks, support vector machines, clustering, dimensionality reduction, kernel methods</li> <li>• fundamentals of statistical learning theory</li> <li>• advanced techniques such as transfer learning, reinforcement learning, generative adversarial networks and adaptive control</li> </ul> <p><i>Skills</i> The students can</p> <ul style="list-style-type: none"> <li>• apply machine learning methods to concrete problems</li> <li>• select and evaluate suitable methods for specific problems</li> <li>• evaluate the quality of a trained data-driven model</li> <li>• work with known software frameworks for machine learning</li> <li>• adapt the architecture and cost function of neural networks to specific problems</li> <li>• show the limits of machine learning methods</li> </ul> <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 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 %	Exercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 min			
<b>Assignment for the Following Curricula</b>	<p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Theoretical Mechanical Engineering: Elective Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Data Science: Compulsory</p> <p>Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory</p> <p>Data Science: Core Qualification: Compulsory</p> <p>Engineering Science: Specialisation Advanced Materials: Elective Compulsory</p> <p>Engineering Science: Specialisation Mechatronics: Elective Compulsory</p> <p>Engineering Science: Specialisation Data Science: Compulsory</p> <p>Engineering Science: Specialisation Mechanical Engineering: Elective Compulsory</p> <p>Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory</p> <p>Logistics and Mobility: Specialisation Information Technology: Elective Compulsory</p> <p>Mechanical Engineering: Specialisation Theoretical Mechanical Engineering: Elective Compulsory</p> <p>Mechatronics: Specialisation Dynamic Systems and AI: Compulsory</p> <p>Technomathematics: Specialisation II. Informatics: Elective Compulsory</p> <p>Engineering and Management - Major in Logistics and Mobility: Specialisation Information Technology: Elective Compulsory</p>			

## Module Manual B.Sc. "Computational Science and Engineering"

Course L2432: Machine Learning I	
<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. Nihat Ay
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>History of neuroscience and machine learning (in particular, the age of deep learning)</li> <li>McCulloch-Pitts neurons and binary Artificial Neural Networks</li> <li>Boolean and threshold functions</li> <li>Universality of McCulloch-Pitts neural networks</li> <li>Learning and the perceptron convergence theorem</li> <li>Support vector machines</li> <li>Harmonic analysis of Boolean functions</li> <li>Continuous Artificial Neural Networks</li> <li>Kolmogorov's superposition theorem</li> <li>Universal approximation with continuous neural networks</li> <li>Approximation error and the gradient decent method: the general idea</li> <li>The stochastic gradient decent method (Robbins-Monro and Kiefer-Wolfowitz cases)</li> <li>Multilayer networks and the backpropagation algorithm</li> <li>Statistical Learning Theory</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>Martin Anthony and Peter L. Bartlett. Neural Network Learning: Theoretical Foundations. Cambridge University Press, 1999.</li> <li>Martin Anthony. Discrete Mathematics of Neural Networks: Selected Topics. SIAM Monographs on Discrete Mathematics &amp; Applications, 1987.</li> <li>Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar. Foundations of Machine Learning, Second Edition. MIT Press, 2018.</li> <li>Christopher M. Bishop. Pattern Recognition and Machine Learning. Information Science and Statistics. Springer-Verlag, 2008.</li> <li>Bernhard Schölkopf, Alexander Smola. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Adaptive Computation and Machine Learning series. MIT Press, Cambridge, MA, 2002.</li> <li>Luc Devroye, László Györfi, Gábor Lugosi. A Probabilistic Theory of Pattern Recognition. Springer, 1996.</li> <li>Vladimir Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag: New York, Berlin, Heidelberg, 1995.</li> </ul>

Course L2433: Machine Learning I	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	3
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 48, Study Time in Lecture 42
<b>Lecturer</b>	Prof. Nihat Ay
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1908: Fundamentals of Operating Systems				
Courses				
Title	Typ		Hrs/wk	CP
Fundamentals of Operating Systems (L3148)	Lecture		2	3
Fundamentals of Operating Systems (L3149)	Recitation Section (small)		2	3
<b>Module Responsible</b>	Prof. Christian Dietrich			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Procedural programming in C, as well as associated tools (editor, linker, compiler)</li> <li>Foundations of computer architecture</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>The course provides basic knowledge about the structure, functionality and system-level use of operating systems. Using the model of a multi-level machine, students learn about operating system abstractions such as processes, threads, virtual memory, files, device files and inter-process communication, as well as techniques for their efficient implementation. This includes strategies for process scheduling, latency minimization through buffering, and main and background memory management. Furthermore, they know the topics of security in the operating system context and aspects of system-oriented software development in C. In the lecture-accompanying exercises, they deepened material practically on the basis programming tasks in C from the range of the UNIX system programming. The students are familiar with the operating system functions for single-processor systems. They have become familiar with special issues relating to multiprocessor systems (based on shared memory) in passing and in relation to functions for coordinating concurrent programs. Similarly, they know the topic of real-time processing to some extent only in relation to process scheduling.</p>			
<i>Skills</i>	<p>Students will be able to use the POSIX system interface to access the various resources of the computing system. They are able to grasp technical documentation in order to implement complex interaction protocols. They are able to recognize concurrency problems and avoid them with blocking synchronization primitives.</p>			
<b>Personal Competence</b> <i>Social Competence</i>	<p>Students are able to discuss and collaboratively present a problem in small groups with reference to operating systems and systems software.</p>			
<i>Autonomy</i>	<p>Students are able to independently prepare and review the lecture content.</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>	<p>General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory            Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory            Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory            Technomathematics: Specialisation II. Informatics: Elective Compulsory</p>			

Course L3148: Fundamentals of Operating 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. Christian Dietrich
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Basic OS concepts</li> <li>• System-oriented software development in C</li> <li>• Files and file systems</li> <li>• Processes and threads</li> <li>• Interrupts, system calls and signals</li> <li>• Process scheduling</li> <li>• Memory based interaction</li> <li>• Resource management, synchronization and jamming</li> <li>• Inter-process communication</li> <li>• Memory organization</li> <li>• Storage virtualization</li> <li>• System security and access protection</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Operating Systems. Internals and Design Principles; William Stallings; Prentice Hall 2008; ISBN: 978-0136006329.</li> <li>• Operating System Concepts; Abraham Silberschatz, Greg Gagne, Peter Bear Galvin; John Wiley &amp; Sons, Inc.; 2005 ISBN: 0-471-69466-5.</li> <li>• Modern Operating Systems; Andrew S. Tanenbaum; Prentice Hall 2007 ISBN: 978-0136006633</li> <li>• Structured Computer Organization; Andrew S. Tanenbaum; Prentice Hall 2006 ISBN: 978-0131485211.</li> </ul>

Course L3149: Fundamentals of Operating 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. Christian Dietrich
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1872: Operating System Construction for Single-Core Systems				
Courses				
Title	Typ		Hrs/wk	CP
Operating System Construction (L2812)	Lecture		2	3
Operating System Construction for Single-Core Systems (L3087)	Project/problem-based Learning		2	3
<b>Module Responsible</b>	Prof. Christian Dietrich			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Object-oriented programming (mandatory)</li> <li>Programming in C/C++ (recommended)</li> <li>Foundations of operating systems (recommended)</li> <li>Foundations of computer architecture (recommended)</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 who have successfully completed the module:</p> <ul style="list-style-type: none"> <li>explain the start-up process of a computing system using an IA32 PC as an example.</li> <li>describe the specific challenges in software development for "bare metal".</li> <li>describe the sequence of an interrupt handling from hardware to (system) software.</li> <li>outline specifics and strategies of interrupt handling in hardware for multi-core systems using the IA32 APIC as an example.</li> <li>distinguish the different types of control flows in an operating system using the level model.</li> <li>distinguish hard, multi-level, and soft methods for interrupt synchronization in operating systems.</li> <li>analyze the interaction of scheduling and interrupt synchronization.</li> <li>distinguish basic ways of coordinating and synchronizing threads (active/passive waiting, non-displaceable critical sections).</li> <li>know basic synchronization problems (lost update, lost wakeup) and propose appropriate countermeasures.</li> <li>can distinguish between different driver models.</li> <li>compare basic OS architectures (library, monolith, microkernel, exokernel, hypervisor) based on fundamental characteristics (robustness, performance, portability) and mechanisms.</li> <li>describe the basic paradigms for interprocess communication in operating systems (memory-based vs. message-based).</li> </ul> <p><i>Skills</i> Students who have successfully completed the module:</p> <ul style="list-style-type: none"> <li>discuss the division of tasks between hardware and system software in interrupt handling.</li> <li>can implement multi-stage interrupt synchronization.</li> <li>classify concrete concurrent situations and derive appropriate synchronization measures.</li> <li>develop the coroutine switch for a given architecture.</li> <li>can implement preemptive scheduling in an operating system.</li> <li>develop mechanisms for thread-level synchronization.</li> <li>can integrate device drivers into an operating system architecture.</li> <li>outline how higher-level synchronization constructs are implemented from basic synchronization primitives (monitors, reader/writer lock).</li> <li>can implement and use primitives for interprocess communication.</li> </ul> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> Students who have successfully completed the module:</p> <ul style="list-style-type: none"> <li>can work cooperatively in small groups.</li> <li>can present and argue their design and implementation decisions in a compact manner.</li> </ul> <p><i>Autonomy</i> Students who have successfully completed the module:</p> <ul style="list-style-type: none"> <li>are able to gradually understand complex error patterns by means of a methodical approach.</li> <li>reflect critically on their decisions and derive alternatives.</li> <li>can deal openly and constructively with weak points and wrong decisions.</li> <li>can revise wrong decisions made or consciously accept the costs incurred.</li> </ul>			
<b>Workload in Hours</b>				
<b>Credit points</b>				
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	No	10 %	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 in Engineering: Specialisation I. Computer Science: 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/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<p>The lecture teaches the conceptual foundations and important techniques required for building an operating system. At the same time, basics from the operating system area such as interrupts, synchronization and scheduling, which should be largely known from other courses, are repeated and deepened.</p> <ul style="list-style-type: none"> <li>• Basics of operating system development</li> <li>• Interrupts (hardware, software, synchronization)</li> <li>• IA-32: The 32-bit Intel architecture</li> <li>• Coroutines and program threads</li> <li>• Scheduling</li> <li>• Operating system architectures</li> <li>• Thread synchronization</li> <li>• Device drivers</li> <li>• Interprocess communication</li> </ul>
<b>Literature</b>	

Course L3087: Operating System Construction for Single-Core Systems	
<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. Christian Dietrich
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<p>The lecture teaches the conceptual foundations and important techniques required for building an operating system. At the same time, basics from the operating system area such as interrupts, synchronization and scheduling, which should be largely known from other courses, are repeated and deepened.</p> <ul style="list-style-type: none"> <li>• Basics of operating system development</li> <li>• Interrupts (hardware, software, synchronization)</li> <li>• IA-32: The 32-bit Intel architecture</li> <li>• Coroutines and program threads</li> <li>• Scheduling</li> <li>• Operating system architectures</li> <li>• Thread synchronization</li> <li>• Device drivers</li> <li>• Interprocess communication</li> </ul> <p>This course deals only with the design of single-core operating systems.</p>
<b>Literature</b>	

## Specialization II. Mathematics & Engineering Science

### Module M1235: Electrical Power Systems I: Introduction to Electrical Power Systems

#### Courses

Title	Typ	Hrs/wk	CP
Electrical Power Systems I: Introduction to Electrical Power Systems (L1670)	Lecture	3	4
Electrical Power Systems I: Introduction to Electrical Power Systems (L1671)	Recitation Section (small)	2	2
<b>Module Responsible</b>	Prof. Christian Becker		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Fundamentals of Electrical Engineering		
<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 give an overview of conventional and modern electric power systems. They can explain in detail and critically evaluate technologies of electric power generation, transmission, storage, and distribution as well as integration of equipment into electric power systems.		
<i>Skills</i>	With completion of this module the students are able to apply the acquired skills in applications of the design, integration, development of electric power systems and to assess the results.		
<b>Personal Competence</b>			
<i>Social Competence</i>	The students can participate in specialized and interdisciplinary discussions, advance ideas and represent their own work results in front of others.		
<i>Autonomy</i>	Students can independently tap knowledge of the emphasis of the lectures.		
<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>	Written exam		
<b>Examination duration and scale</b>	90 - 150 minutes		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Electrical Engineering: Elective Compulsory General Engineering Science (German program, 7 semester): Specialisation Green Technologies, Focus Renewable Energy: Elective Compulsory Data Science: Core Qualification: Elective Compulsory Electrical Engineering: Core Qualification: Elective Compulsory Energy Systems: Specialisation Energy Systems: Elective Compulsory Engineering Science: Specialisation Electrical Engineering: Elective Compulsory Green Technologies: Energy, Water, Climate: Specialisation Energy Systems / Renewable Energies: Elective Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Integrated Building Technology: Core Qualification: Compulsory Mechatronics: Specialisation Electrical Systems: Elective Compulsory Renewable Energies: Core Qualification: Compulsory Theoretical Mechanical Engineering: Specialisation Energy Systems: Elective Compulsory		

Course L1670: Electrical Power Systems I: Introduction to Electrical Power 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. Christian Becker
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• fundamentals and current development trends in electric power engineering</li> <li>• tasks and history of electric power systems</li> <li>• symmetric three-phase systems</li> <li>• fundamentals and modelling of electric power systems <ul style="list-style-type: none"> <li>◦ lines</li> <li>◦ transformers</li> <li>◦ synchronous machines</li> <li>◦ induction machines</li> <li>◦ loads and compensation</li> <li>◦ grid structures and substations</li> </ul> </li> <li>• fundamentals of energy conversion <ul style="list-style-type: none"> <li>◦ electro-mechanical energy conversion</li> <li>◦ thermodynamics</li> <li>◦ power station technology</li> <li>◦ renewable energy conversion systems</li> </ul> </li> <li>• steady-state network calculation <ul style="list-style-type: none"> <li>◦ network modelling</li> <li>◦ load flow calculation</li> <li>◦ (n-1)-criterion</li> </ul> </li> <li>• symmetric failure calculations, short-circuit power</li> <li>• control in networks and power stations</li> <li>• grid protection</li> <li>• grid planning</li> <li>• power economy fundamentals</li> </ul>
<b>Literature</b>	<p>K. Heuck, K.-D. Dettmann, D. Schulz: "Elektrische Energieversorgung", Vieweg + Teubner, 9. Auflage, 2013</p> <p>A. J. Schwab: "Elektroenergiesysteme", Springer, 5. Auflage, 2017</p> <p>R. Flösdorff: "Elektrische Energieverteilung" Vieweg + Teubner, 9. Auflage, 2008</p>

Course L1671: Electrical Power Systems I: Introduction to Electrical Power Systems	
<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>	Prof. Christian Becker
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• fundamentals and current development trends in electric power engineering</li> <li>• tasks and history of electric power systems</li> <li>• symmetric three-phase systems</li> <li>• fundamentals and modelling of electric power systems <ul style="list-style-type: none"> <li>◦ lines</li> <li>◦ transformers</li> <li>◦ synchronous machines</li> <li>◦ induction machines</li> <li>◦ loads and compensation</li> <li>◦ grid structures and substations</li> </ul> </li> <li>• fundamentals of energy conversion <ul style="list-style-type: none"> <li>◦ electro-mechanical energy conversion</li> <li>◦ thermodynamics</li> <li>◦ power station technology</li> <li>◦ renewable energy conversion systems</li> </ul> </li> <li>• steady-state network calculation <ul style="list-style-type: none"> <li>◦ network modelling</li> <li>◦ load flow calculation</li> <li>◦ (n-1)-criterion</li> </ul> </li> <li>• symmetric failure calculations, short-circuit power</li> <li>• control in networks and power stations</li> <li>• grid protection</li> <li>• grid planning</li> <li>• power economy fundamentals</li> </ul>
<b>Literature</b>	<p>K. Heuck, K.-D. Dettmann, D. Schulz: "Elektrische Energieversorgung", Vieweg + Teubner, 9. Auflage, 2013</p> <p>A. J. Schwab: "Elektroenergiesysteme", Springer, 5. Auflage, 2017</p> <p>R. Flösdorff: "Elektrische Energieverteilung" Vieweg + Teubner, 9. Auflage, 2008</p>

Module M0760: Electronic Devices				
Courses				
Title	Typ		Hrs/wk	CP
Electronic Devices (L0720)	Lecture		3	4
Electronic Devices (L0721)	Project-/problem-based Learning		2	2
<b>Module Responsible</b>	Prof. Hoc Khiem Trieu			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Atomic model and quantum theory, electrical currents in solid state materials, basics in solid-state physics Successful participation of Physics for Engineers and Materials in Electrical Engineering or courses with equivalent contents			
<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 <ul style="list-style-type: none"> <li>to represent the basics of semiconductor physics,</li> <li>to explain the operating principle of important semiconductor devices,</li> <li>to outline device characteristics and equivalent circuits as well as to explain their derivation and</li> <li>to discuss the limitation of device models.</li> </ul>			
<i>Skills</i>	Students are capable <ul style="list-style-type: none"> <li>to apply devices in basic circuits,</li> <li>to realize the physical context and to solve complex problems by oneself</li> </ul>			
<b>Personal Competence</b> <i>Social Competence</i>	Students are able to prepare and perform their lab experiments in team work as well as to present and discuss the results in front of audience.			
<i>Autonomy</i>	Students are capable to acquire knowledge based on literature in order to prepare their experiments.			
<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> Yes	<b>Bonus</b> 10 %	<b>Form</b> Subject theoretical and practical work	<b>Description</b> Studierenden erarbeiten in Kleingruppen Wissen zu einem bestimmten Thema, demonstrieren dieses in Form eines Versuches mit Präsentation und Diskussion. Darüber hinaus betreut jede Gruppe eine Übungsaufgabe, die inhaltlich zu dem jeweiligen Versuch gehört.
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	120 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Electrical Engineering: Compulsory Electrical Engineering: Core Qualification: Compulsory Engineering Science: Specialisation Electrical Engineering: Compulsory General Engineering Science (English program, 7 semester): Specialisation Electrical Engineering: Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Mechatronics: Specialisation Electrical Systems: Compulsory			

Course L0720: Electronic Devices	
<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. Hoc Khiem Trieu
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>Uniformly doped semiconductor (semiconductor, crystal structure, energy band diagram, effective mass, density of state, probability of occupancy, mass action law, generation and recombination processes, generation and recombination lifetime, carrier transport mechanisms: drift current, diffusion current; equilibriums in semiconductor, semiconductor equations)</li> <li>pn-junction (zero applied bias, energy band diagram in thermal equilibrium, current-voltage characteristics, derivation of diode equation, consideration of space charge recombination, transient behaviour, breakdown mechanisms, various types of diodes: Zener diode, tunnel diode, backward diode, photo diode, LED, laser diode)</li> <li>Bipolar transistor (principle of operation, current-voltage characteristics: calculation of base, collector and emitter current, operating modes; non-ideality: actual doping profile, Early effect, breakdown, generation and recombination current and high injection; Ebers-Moll model: family of characteristics, equivalent circuit; frequency response, switching characteristics, heterojunction bipolar transistor)</li> <li>Unipolar devices (surface effects: surface states, work function, energy band diagram; metal-semiconductor junctions: Schottky contact, current-voltage characteristics, ohmic contact; junction field effect transistor: operating principle, current-voltage characteristics, small-signal model, breakdown characteristics; MESFET: operating principle, depletion mode and enhancement mode MESFET; MIS structure: accumulation, depletion, inversion, strong inversion, flatband voltage, oxide charges, threshold voltage, capacitance voltage characteristics; MOSFET: basic structure, principle of operation, current voltage characteristics, frequency response, subthreshold behaviour, threshold voltage, device scaling; CMOS)</li> </ul>
<b>Literature</b>	<p>S.M. Sze: Semiconductor devices, Physics and Technology, John Wiley &amp; Sons (1985) F. Thuselt: Physik der Halbleiterbauelemente, Springer (2011)</p> <p>T. Thille, D. Schmitt-Landsiedel: Mikroelektronik, Halbleiterbauelemente und deren Anwendung in elektronischen Schaltungen, Springer (2004)</p> <p>B.L. Anderson, R.L. Anderson: Fundamentals of Semiconductor Devices, McGraw-Hill (2005)</p> <p>D.A. Neamen: Semiconductor Physics and Devices, McGraw-Hill (2011)</p> <p>M. Shur: Introduction to Electronic Devices, John Wiley &amp; Sons (1996)</p> <p>S.M. Sze: Physics of semiconductor devices, John Wiley &amp; Sons (2007)</p> <p>H. Schaumburg: Halbleiter, B.G. Teubner (1991)</p> <p>A. Möschwitzer: Grundlagen der Halbleiter-&amp;Mikroelektronik, Bd1 Elektronische Halbleiterbauelemente, Carl Hanser (1992)</p> <p>H.-G. Unger, W. Schultz, G. Weinhausen: Elektronische Bauelemente und Netzwerke I, Physikalische Grundlagen der Halbleiterbauelemente, Vieweg (1985)</p>

Course L0721: Electronic Devices	
<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. Hoc Khiem Trieu
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1896: Machine Dynamics				
Courses				
Title	Typ		Hrs/wk	CP
Machine Dynamics (L3144)	Lecture		3	3
Machine Dynamics (L3145)	Project-/problem-based Learning		3	3
<b>Module Responsible</b>	Dr. Alireza Abbasimoshaei			
<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 96, Study Time in Lecture 84			
<b>Credit points</b>	6			
<b>Course achievement</b>	None			
<b>Examination</b>	Subject theoretical and practical work			
<b>Examination duration and scale</b>	70% written exam (120 minutes) duration and 30% project			
<b>Assignment for the Following Curricula</b>	Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Mechatronics: Core Qualification: Elective Compulsory			

Course L3144: Machine Dynamics	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 48, Study Time in Lecture 42
<b>Lecturer</b>	Dr. Alireza Abbasimoshaei
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<p><b>1: Mechanisms</b></p> <p>1.1 Introduction</p> <p>1.2 Types of Kinematic Joints</p> <p>1.3 Elements Or Links</p> <p>1.4 Constrained Motion</p> <p>1.6 Kinematic Chain</p> <p>1.7 Types of Mechanisms and Equivalent Mechanisms</p> <p>1.8 Classification of Machines</p> <p>1.9 Degrees of Freedom</p> <p>1.10 Four-Bar Chain</p> <p>1.11 Grashof's and Grubler's Law</p> <p>1.12 Inversion of Mechanisms</p> <p>1.13 Simulation in software</p> <p><b>2: Velocity in Mechanisms</b></p> <p>2.1 Introduction</p> <p>2.2 Velocity Diagrams</p> <p>2.3 Determination of Link Velocities</p> <p>2.4 Relative Velocity (linear and angular)</p> <p>2.5 Instantaneous Centre Method and its types</p> <p>2.6 Analyses in Software</p> <p><b>3: Acceleration in Mechanisms</b></p> <p>3.1 Introduction</p> <p>3.2 Acceleration of a Body Moving in a Circular Path</p> <p>3.3 Acceleration Diagrams and Center for Different Mechanisms</p> <p>3.4 Coriolis Acceleration</p> <p>3.5 Link Sliding Acceleration</p> <p>3.7 Analytical Analysis of Different Mechanisms Properties in Software</p> <p><b>4: Belts, Chains, Ropes, Clutches, and Brakes</b></p> <p>4.1 Introduction</p> <p>4.2 Flat Belt Drive and Velocity and Tension Ratio</p> <p>4.3 V-Belt Drive</p>

4.4 Chain Drive and Pitch

4.5 Rope Drive

4.6 Types of Brakes and their analyses

4.7 Types of Clutches and their analyses

4.8 Driving their Equations in Software

#### **5: Cams**

5.1 Introduction

5.2 Classification of Cams

5.3 Types of Followers

5.4 Cam Profile

5.5 Follower Different Motions

5.6 Cam Profile with Knife-Edge Follower

5.7 Cam Profile with Roller Follower

5.8 Cam Profile with Translational Flat-Faced Follower

5.9 Cam Profile with Swinging Roller Follower

5.10 Analytical Methods

5.11 Radius of Curvature and Undercutting

5.12 Cam Size

5.13 Initial Design of a Cam and its Profile Driving by Software

#### **6: Static and Dynamic Force Analysis**

6.1 Introduction

6.2 Static Force Analysis and Equilibrium

6.3 Dynamic Force Analysis

6.4 Force Convention and Free Body Diagrams

6.5 Principle of Superposition

6.6 Force Analyses in Softwares and drive the equations

#### **7: Balancing**

7.1 Introduction

7.2 Balancing of Rotating Masses and Analytical Method for Balancing

7.3 Reciprocating Masses

7.4 Reciprocating Engine

7.5 Primary Balance

7.6 Multicylinder In-Line Engines

7.7 Secondary Balancing

7.8 Balancing of Radial Engines, V-Engines, and Rotors

7.9 Static Balance

7.10 Dynamic Balance

7.11 Flexible Rotor Balancing

7.12 Balancing Machines

7.13 Balancing Analyse in Software

#### **8: Gyroscopic and Precessional Motion**

8.1 Introduction

8.2 Precessional Motion

8.3 Fundamentals of Gyroscopic Motion

8.4 Gyroscopic Couple of a Plane Disc

8.5 Effect of Gyroscopic Couple on Bearings

8.6 Gyroscopic Couple on an Aeroplane

8.7 Stability of a Two and Four-Wheel Vehicle Taking a Turn

8.8 Effect of Precession on a Disc Fixed at a Certain Angle to a Rotating Shaft

8.9 Gyroscopic Analysis in Software

#### **9: Gear Trains**

9.1 Introduction

9.2 Types of Gear Trains

9.3 Determination of Speed Ratio of Planetary Gear Trains

9.4 Sun and Planet Gears and Their equations

9.5 Epicyclics with Two Inputs

9.6 Compound Epicyclic Gear Train

9.7 Epicyclic Bevel Gear Trains

9.8 Torque in Epicyclic Gear Trains

9.9 Gear Movement analyses in Software

#### **10: Kinematic Synthesis of Planar Mechanisms**

10.1 Introduction

10.2 Movability (or Mobility) or Number Synthesis

## Module Manual B.Sc. "Computational Science and Engineering"

	<p>10.3 Transmission Angle in Different Mechanisms</p> <p>10.4 Limit Positions and Dead Centres of a Four-Bar Mechanism</p> <p>10.5 Dimensional Synthesis</p> <p>10.6 Graphical Method of Synthesis</p> <p>10.7 Design of Different Mechanisms by Relative Pole Method</p> <p>10.8 Errors in Kinematic Synthesis of Mechanisms</p> <p>10.9 Analytical Method (Function Generation, Chebyshev's Spacing, Freudenstein's Equation)</p> <p>10.10 Implementing Synthesis Methods in Softwares</p> <p><b>11: Mechanical Vibrations</b></p> <p>11.1 Introduction</p> <p>11.2 Definitions</p> <p>11.3 Types of Free Vibrations</p> <p>11.4 Basic Elements of Vibrating System</p> <p>11.5 Degrees of Freedom</p> <p>11.6 Simple Harmonic Motion</p> <p>11.7 Free Longitudinal Vibrations</p> <p>11.8 Effect of the Spring Mass and Equivalent Stiffness</p> <p>11.9 Critical Speed</p> <p>11.10 Geared System</p>
<b>Literature</b>	<p>1. Mechanisms and Machines: Kinematics, Dynamics, and Synthesis: Michael M Stanisic</p> <p>2. Kinematics and Dynamics of Machines: George H. Martin</p> <p>3. Machine Dynamics in Mechatronic Systems an engineering approach: Adrian M. Rankers</p>

Course L3145: Machine Dynamics	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	3
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 48, Study Time in Lecture 42
<b>Lecturer</b>	Dr. Alireza Abbasimoshaei
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0708: Electrical Engineering III: Circuit Theory and Transients				
Courses				
Title	Typ		Hrs/wk	CP
Circuit Theory (L0566)	Lecture		3	4
Circuit Theory (L0567)	Recitation Section (small)		2	2
<b>Module Responsible</b>	Prof. Alexander Kölpin			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Electrical Engineering I and II, Mathematics I and II			
<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 explain the basic methods for calculating electrical circuits. They know the Fourier series analysis of linear networks driven by periodic signals. They know the methods for transient analysis of linear networks in time and in frequency domain, and they are able to explain the frequency behaviour and the synthesis of passive two-terminal-circuits.			
<i>Skills</i>	The students are able to calculate currents and voltages in linear networks by means of basic methods, also when driven by periodic signals. They are able to calculate transients in electrical circuits in time and frequency domain and are able to explain the respective transient behaviour. They are able to analyse and to synthesize the frequency behaviour of passive two-terminal-circuits.			
<b>Personal Competence</b>				
<i>Social Competence</i>	Students work on exercise tasks in small guided groups. They are encouraged to present and discuss their results within the group.			
<i>Autonomy</i>	The students are able to find out the required methods for solving the given practice problems. Possibilities are given to test their knowledge during the lectures continuously by means of short-time tests. This allows them to control independently their educational objectives. They can link their gained knowledge to other courses like Electrical Engineering I and Mathematics I.			
<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>	Written exam			
<b>Examination duration and scale</b>	150 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Mechatronics: Compulsory General Engineering Science (German program, 7 semester): Specialisation Electrical Engineering: Compulsory Electrical Engineering: Core Qualification: Compulsory Engineering Science: Specialisation Electrical Engineering: Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Mechatronics: Specialisation Electrical Systems: Compulsory Mechatronics: Specialisation Dynamic Systems and AI: Compulsory Mechatronics: Core Qualification: Compulsory Mechatronics: Specialisation Robot- and Machine-Systems: Compulsory Technomathematics: Specialisation III. Engineering Science: Elective Compulsory			

Course L0566: Circuit 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. Alexander Kölpin, Dr. Fabian Lurz
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>- Circuit theorems</li> <li>- N-port circuits</li> <li>- Periodic excitation of linear circuits</li> <li>- Transient analysis in time domain</li> <li>- Transient analysis in frequency domain; Laplace Transform</li> <li>- Frequency behaviour of passive one-ports</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>- M. Albach, "Grundlagen der Elektrotechnik 1", Pearson Studium (2011)</li> <li>- M. Albach, "Grundlagen der Elektrotechnik 2", Pearson Studium (2011)</li> <li>- L. P. Schmidt, G. Schaller, S. Martius, "Grundlagen der Elektrotechnik 3", Pearson Studium (2011)</li> <li>- T. Harriehausen, D. Schwarzenau, "Moeller Grundlagen der Elektrotechnik", Springer (2013)</li> <li>- A. Hambley, "Electrical Engineering: Principles and Applications", Pearson (2008)</li> <li>- R. C. Dorf, J. A. Svoboda, "Introduction to electrical circuits", Wiley (2006)</li> <li>- L. Moura, I. Darwazeh, "Introduction to Linear Circuit Analysis and Modeling", Amsterdam Newnes (2005)</li> </ul>

Course L0567: Circuit Theory	
<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>	Prof. Alexander Kölpin, Dr. Fabian Lurz
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	see interlocking course
<b>Literature</b>	siehe korrespondierende Lehrveranstaltung

Module M0941: Combinatorial Structures and Algorithms			
Courses			
Title	Typ	Hrs/wk	CP
Combinatorial Structures and Algorithms (L1100)	Lecture	3	4
Combinatorial Structures and Algorithms (L1101)	Recitation Section (small)	1	2
<b>Module Responsible</b>	Prof. Anusch Taraz		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Mathematics I + II</li> <li>Discrete Algebraic Structures</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> <i>Knowledge</i>	<ul style="list-style-type: none"> <li>Students can name the basic concepts in Combinatorics and Algorithms. 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>Skills</i>	<ul style="list-style-type: none"> <li>Students can model problems in Combinatorics and Algorithms 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> <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>	Oral exam		
<b>Examination duration and scale</b>	30 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Core Qualification: Elective Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory		

Course L1100: Combinatorial Structures and Algorithms	
<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. Anusch Taraz, Dr. Dennis Clemens
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Counting</li> <li>• Structural Graph Theory</li> <li>• Analysis of Algorithms</li> <li>• Extremal Combinatorics</li> <li>• Random discrete structures</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• M. Aigner: Diskrete Mathematik, Vieweg, 6. Aufl., 2006</li> <li>• J. Matoušek &amp; J. Nešetřil: Diskrete Mathematik - Eine Entdeckungsreise, Springer, 2007</li> <li>• A. Steger: Diskrete Strukturen - Band 1: Kombinatorik, Graphentheorie, Algebra, Springer, 2. Aufl. 2007</li> <li>• A. Taraz: Diskrete Mathematik, Birkhäuser, 2012.</li> </ul>

Course L1101: Combinatorial Structures and Algorithms	
<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. Anusch Taraz
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1802: Engineering Mechanics I (Stereostatics)				
Courses				
Title		Type	Hrs/wk	CP
Engineering Mechanics I (Statics) (L1001)		Lecture	2	3
Engineering Mechanics I (Statics) (L1003)		Recitation Section (large)	1	1
Engineering Mechanics I (Statics) (L1002)		Recitation Section (small)	2	2
Module Responsible	Prof. Benedikt Kriegesmann			
Admission Requirements	None			
Recommended Previous Knowledge	Solid school knowledge in mathematics and physics.			
Educational Objectives	After taking part successfully, students have reached the following learning results			
Professional Competence	Knowledge	The students can		
		<ul style="list-style-type: none"><li>describe the axiomatic procedure used in mechanical contexts;</li><li>explain important steps in model design;</li><li>present technical knowledge in stereostatics.</li></ul>		
	Skills	The students can		
		<ul style="list-style-type: none"><li>explain the important elements of mathematical / mechanical analysis and model formation, and apply it to the context of their own problems;</li><li>apply basic statical methods to engineering problems;</li><li>estimate the reach and boundaries of statical methods and extend them to be applicable to wider problem sets.</li></ul>		
Personal Competence				
Social Competence	The students can work in groups and support each other to overcome difficulties.			
Autonomy	Students are capable of determining their own strengths and weaknesses and to organize their time and learning based on those.			
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70			
Credit points	6			
Course achievement	None			
Examination	Written exam			
Examination duration and scale	90 min			
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Core Qualification: Compulsory Civil- and Environmental Engineering: Core Qualification: Compulsory Bioprocess Engineering: Core Qualification: Compulsory Chemical and Bioprocess Engineering: Core Qualification: Compulsory Data Science: Specialisation II. Application: Elective Compulsory Electrical Engineering: Core Qualification: Elective Compulsory Green Technologies: Energy, Water, Climate: Core Qualification: Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Integrated Building Technology: Core Qualification: Compulsory Mechanical Engineering: Core Qualification: Compulsory Mechatronics: Core Qualification: Compulsory Orientation Studies: Core Qualification: Elective Compulsory Naval Architecture: Core Qualification: Compulsory Process Engineering: Core Qualification: Compulsory Engineering and Management - Major in Logistics and Mobility: Core Qualification: Compulsory			

Course L1001: Engineering Mechanics I (Statics)	
<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. Benedikt Kriegesmann
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Tasks in Mechanics</li> <li>• Modelling and model elements</li> <li>• Vector calculus for forces and torques</li> <li>• Forces and equilibrium in space</li> <li>• Constraints and reactions, characterization of constraint systems</li> <li>• Planar and spatial truss structures</li> <li>• Internal forces and moments for beams and frames</li> <li>• Center of mass, volume, area and line</li> <li>• Computation of center of mass by integrals, joint bodies</li> <li>• Friction (sliding and sticking)</li> <li>• Friction of ropes</li> </ul>
<b>Literature</b>	<b>K. Magnus, H.H. Müller-Slany: Grundlagen der Technischen Mechanik. 7. Auflage, Teubner (2009).</b> <b>D. Gross, W. Hauger, J. Schröder, W. Wall: Technische Mechanik 1. 11. Auflage, Springer (2011).</b>

Course L1003: Engineering Mechanics I (Statics)	
<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. Benedikt Kriegesmann
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	Forces and equilibrium Constraints and reactions Frames Center of mass Friction Internal forces and moments for beams
<b>Literature</b>	K. Magnus, H.H. Müller-Slany: Grundlagen der Technischen Mechanik. 7. Auflage, Teubner (2009). D. Gross, W. Hauger, J. Schröder, W. Wall: Technische Mechanik 1. 11. Auflage, Springer (2011).

Course L1002: Engineering Mechanics I (Statics)	
<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>	Prof. Benedikt Kriegesmann
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	Forces and equilibrium Constraints and reactions Frames Center of mass Friction Internal forces and moments for beams
<b>Literature</b>	K. Magnus, H.H. Müller-Slany: Grundlagen der Technischen Mechanik. 7. Auflage, Teubner (2009). D. Gross, W. Hauger, J. Schröder, W. Wall: Technische Mechanik 1. 11. Auflage, Springer (2011).

Module M0783: Measurements: Methods and Data Processing				
Courses				
Title	Typ		Hrs/wk	CP
EE Experimental Lab (L0781)	Practical Course		2	2
Measurements: Methods and Data Processing (L0779)	Lecture		2	3
Measurements: Methods and Data Processing (L0780)	Recitation Section (small)		1	1
<b>Module Responsible</b>	Prof. Alexander Schläefer			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	principles of mathematics principles of electrical engineering			
<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 explain the purpose of metrology and the acquisition and processing of measurements. They can detail aspects of probability theory and errors, and explain the processing of stochastic signals. Students know methods to digitalize and describe measured signals.			
<i>Skills</i>	The students are able to evaluate problems of metrology and to apply methods for describing and processing of measurements.			
<b>Personal Competence</b>				
<i>Social Competence</i>	The students solve problems in small groups.			
<i>Autonomy</i>	The students can reflect their knowledge and discuss and evaluate their results.			
<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 %	Exercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Electrical Engineering: Elective Compulsory Electrical Engineering: Core Qualification: Compulsory Engineering Science: Specialisation Electrical Engineering: Elective Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Integrated Building Technology: Core Qualification: Elective Compulsory Technomathematics: Specialisation III. Engineering Science: Elective Compulsory			

Course L0781: EE Experimental Lab	
<b>Typ</b>	Practical Course
<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 Schläefer, Dozenten des SD E, Prof. Alexander Kölpin, Prof. Bernd-Christian Renner, Prof. Christian Becker, Prof. Heiko Falk, Prof. Herbert Werner, Prof. Thorsten Kern
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	lab experiments: digital circuits, semiconductors, micro controllers, analog circuits, AC power, electrical machines
<b>Literature</b>	Wird in der Lehrveranstaltung festgelegt

Course L0779: Measurements: Methods and Data Processing	
<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 Schläefer
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	introduction, systems and errors in metrology, probability theory, measuring stochastic signals, describing measurements, acquisition of analog signals, applied metrology
<b>Literature</b>	Puente León, Kiencke: Messtechnik, Springer 2012 Lerch: Elektrische Messtechnik, Springer 2012  Weitere Literatur wird in der Veranstaltung bekanntgegeben.

Course L0780: Measurements: Methods and Data Processing	
<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>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1712: Green Technologies II			
Courses			
Title	Typ	Hrs/wk	CP
Practical Exercise Environmental Technology (L1387)	Practical Course	1	1
Pollutant analysis (L2996)	Lecture	2	3
Environmental Technologie (L0326)	Lecture	2	2
<b>Module Responsible</b>	Dr. Marvin Scherzinger		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Fundamentals of inorganic/organic chemistry and biology.		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	<p>With the completion of this modul the students obtain profound knowledge of environmental technology. They are able to describe the behaviour of chemicals in the environment. Students can give an overview of scientific disciplines involved. They can explain terms and allocate them to related methods.</p> <p>Additional students acquire in-depth knowledge of important cause-effect chains of potential environmental problems which might occur from production processes, projects or construction measures. They have knowledge about the methodological diversity and are competent in dealing with different methods and instruments to assess environmental impacts. Besides the students are able to estimate the complexity of these environmental processes as well as uncertainties and difficulties with their measurement.</p>		
<i>Skills</i>	<p>Students are able to propose appropriate management and mitigation measures for environmental problems. They are able to determine geochemical parameters and to assess the potential of pollutants to migrate and transform. The students are able to work out well founded opinions on how Environmental Technology contributes to sustainable development, and they can present and defend these opinions in front of and against the group.</p> <p>The students are able to select a suitable method for the respective case from the variety of assessment methods. Thereby they can develop suitable solutions for managing and mitigating environmental problems in a business context. They are able to carry out Life Cycle Impact Assessments independently and can apply the software programs OpenLCA and the database EcoInvent. After finishing the course the students have the competence to critically judge research results or other publications on environmental impacts.</p>		
<b>Personal Competence</b>			
<i>Social Competence</i>	<p>The students are able to discuss the various technical and scientific tasks, both subject-specific and multidisciplinary. They are able to develop different approaches to the task as a group as well as to discuss their theoretical or practical implementation.</p> <p>Due to the selected lecture topics, the students receive insights into the multi-layered issues of the environment protection and the concept of sustainability. Their sensitivity and consciousness towards these subjects are raised and which helps to raise their awareness of their future social responsibilities in their role as engineers.</p>		
<i>Autonomy</i>	<p>The students learn to research, process and present a scientific topic independently. They are able to carry out independent scientific work. They can solve an environmental problem in a business context and are able to judge results of other publications.</p>		
<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>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	<p>General Engineering Science (German program, 7 semester): Specialisation Green Technologies: Compulsory</p> <p>Green Technologies: Energy, Water, Climate: Core Qualification: Compulsory</p> <p>Computer Science in Engineering: Specialisation II. Mathematics &amp; Engineering Science: Elective Compulsory</p>		

Course L1387: Practical Exercise Environmental Technology	
<b>Typ</b>	Practical Course
<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. Martin Kaltschmitt, Dr. Marvin Scherzinger
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<p>The practical course Environmental Engineering currently consists of 5 experiments, which deal with the different focal points of environmental engineering in the areas of air, water, soil, energy and noise. The following experiments are carried out for this purpose:</p> <p>biological degradation of artificial materials,</p> <p>fine dust measurement in the air,</p> <p>water analysis,</p> <p>noise emission measurement,</p> <p>photovoltaic energy</p> <p>Within the lab course students discuss the various technical and scientific tasks, both subject-specific and multidisciplinary. They discuss different approaches to the task as well as it's theoretical or practical implementation.</p>
<b>Literature</b>	Folien der Einführungsveranstaltung

Course L2996: Pollutant analysis	
<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. Marvin Scherzinger
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<p>In this course, modern analytical methods are presented that are used for the quantification of pollutants in the environmental compartments soil, water and air. In doing so, the students deepen their theoretical knowledge with regard to working with standardized methods and learn to make statements about the quality of test results.</p>
<b>Literature</b>	Vorlesungsfolien

Course L0326: Environmental Technologie	
<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. Martin Kaltschmitt, Dr. Marvin Scherzinger
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<ol style="list-style-type: none"> <li>1. Introductory seminar on environmental science:</li> <li>2. Environmental impact and adverse effects</li> <li>3. Wastewater technology</li> <li>4. Air pollution control</li> <li>5. Noise protection</li> <li>6. Waste and recycling management</li> <li>7. Soil and ground water protection</li> <li>8. Renewable energies</li> <li>9. Resource conservation and energy efficiency</li> </ol>
<b>Literature</b>	Förster, U.: Umweltschutztechnik; 2012; Springer Berlin (Verlag) 8., Aufl. 2012; 978-3-642-22972-5 (ISBN)

Module M0634: Introduction into Medical Technology and Systems				
Courses				
Title	Typ		Hrs/wk	CP
Introduction into Medical Technology and Systems (L0342)	Lecture		2	3
Introduction into Medical Technology and Systems (L0343)	Project Seminar		2	2
Introduction into Medical Technology and Systems (L1876)	Recitation Section (large)		1	1
<b>Module Responsible</b>	Prof. Alexander Schläefer			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	principles of math (algebra, analysis/calculus) principles of stochastics principles of programming, R/Matlab			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	The students can explain principles of medical technology, including imaging systems, computer aided surgery, and medical information systems. They are able to give an overview of regulatory affairs and standards in medical technology.  The students are able to evaluate systems and medical devices in the context of clinical applications.  The students describe a problem in medical technology as a project, and define tasks that are solved in a joint effort. The students can critically reflect on the results of other groups and make constructive suggestions for improvement.  The students can assess their level of knowledge and document their work results. They can critically evaluate the results achieved and present them in an appropriate manner.			
<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>
	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>	General Engineering Science (German program, 7 semester): Specialisation Biomedical Engineering: Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Specialisation II. Application: Elective Compulsory Data Science: Core Qualification: Elective Compulsory Electrical Engineering: Core Qualification: Elective Compulsory Engineering Science: Specialisation Biomedical Engineering: Compulsory General Engineering Science (English program, 7 semester): Specialisation Biomedical Engineering: Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Mechatronics: Specialisation Medical Engineering: 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 Technomathematics: Specialisation III. Engineering Science: Elective Compulsory			

Course L0342: Introduction into Medical Technology and 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. Alexander Schlaefer
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>- imaging systems</li> <li>- computer aided surgery</li> <li>- medical sensor systems</li> <li>- medical information systems</li> <li>- regulatory affairs</li> <li>- standard in medical technology</li> </ul> <p>The students will work in groups to apply the methods introduced during the lecture using problem based learning.</p>
<b>Literature</b>	<p>Bernhard Priem, "Visual Computing for Medicine", 2014</p> <p>Heinz Handels, "Medizinische Bildverarbeitung", 2009 (<a href="https://katalog.tub.tuhh.de/Record/745558097">https://katalog.tub.tuhh.de/Record/745558097</a>)</p> <p>Valery Tuchin, "Tissue Optics - Light Scattering Methods and Instruments for Medical Diagnosis", 2015</p> <p>Olaf Drössel, "Biomedizinische Technik - Medizinische Bildgebung", 2014</p> <p>H. Gross, "Handbook of Optical Systems", 2008 (<a href="https://katalog.tub.tuhh.de/Record/856571687">https://katalog.tub.tuhh.de/Record/856571687</a>)</p> <p>Wolfgang Drexler, "Optical Coherence Tomography", 2008</p> <p>Kramme, "Medizintechnik", 2011</p> <p>Thorsten M. Buzug, "Computed Tomography", 2008</p> <p>Otmar Scherzer, "Handbook of Mathematical Methods in Imaging", 2015</p> <p>Weishaupt, "Wie funktioniert MRI?", 2014</p> <p>Paul Suetens, "Fundamentals of Medical Imaging", 2009</p> <p>Vorlesungsunterlagen</p>

Course L0343: Introduction into Medical Technology and Systems	
<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>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L1876: Introduction into Medical Technology and Systems	
<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. Alexander Schlaefer
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0715: Solvers for Sparse Linear Systems				
Courses				
Title		Typ	Hrs/wk	CP
Solvers for Sparse Linear Systems (L0583)		Lecture	2	3
Solvers for Sparse Linear Systems (L0584)		Recitation Section (small)	2	3
Module Responsible		Prof. Sabine Le Borne		
Admission Requirements		None		
Recommended Previous Knowledge		<ul style="list-style-type: none"><li>Mathematics I + II for Engineering students or Analysis &amp; Lineare Algebra I + II for Technomathematicians</li><li>Programming experience in C</li></ul>		
Educational Objectives		After taking part successfully, students have reached the following learning results		
Professional Competence		<div><div>Knowledge</div><div>Students can<ul style="list-style-type: none"><li>list classical and modern iteration methods and their interrelationships,</li><li>repeat convergence statements for iterative methods,</li><li>explain aspects regarding the efficient implementation of iteration methods.</li></ul></div><div><div>Skills</div><div>Students are able to<ul style="list-style-type: none"><li>analyse, implement, test, and compare iterative methods,</li><li>analyse the convergence behaviour of iterative methods and, if applicable, compute congerence rates.</li></ul></div></div></div>		
Personal Competence				
Social Competence				
Autonomy				
Workload in Hours		Independent Study Time 124, Study Time in Lecture 56		
Credit points		6		
Course achievement		None		
Examination		Oral exam		
Examination duration and scale		20 min		
Assignment for the Following Curricula		Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Core Qualification: Elective Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory		

Course L0583: Solvers for Sparse Linear 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. Sabine Le Borne
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	1. Sparse systems: Orderings and storage formats, direct solvers 2. Classical methods: basic notions, convergence 3. Projection methods 4. Krylov space methods 5. Preconditioning (e.g. ILU) 6. Multigrid methods 7. Domain Decomposition Methods
<b>Literature</b>	1. Y. Saad. Iterative methods for sparse linear systems 2. M. Olshanskii, E. Tyrtyshnikov. Iterative methods for linear systems: theory and applications

Course L0584: Solvers for Sparse Linear 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. Sabine Le Borne
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0777: Semiconductor Circuit Design			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Semiconductor Circuit Design (L0763)	Lecture	3	4
Semiconductor Circuit Design (L0864)	Recitation Section (small)	1	2
<b>Module Responsible</b>	NN		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Fundamentals of electrical engineering Basics of physics, especially semiconductor physics		
<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 explain the functionality of different MOS devices in electronic circuits.</li> <li>Students are able to explain how analog circuits functions and where they are applied.</li> <li>Students are able to explain the functionality of fundamental operational amplifiers and their specifications.</li> <li>Students know the fundamental digital logic circuits and can discuss their advantages and disadvantages.</li> <li>Students have knowledge about memory circuits and can explain their functionality and specifications.</li> <li>Students know the appropriate fields for the use of bipolar transistors.</li> </ul>		
<i>Skills</i>	<ul style="list-style-type: none"> <li>Students can calculate the specifications of different MOS devices and can define the parameters of electronic circuits.</li> <li>Students are able to develop different logic circuits and can design different types of logic circuits.</li> <li>Students can use MOS devices, operational amplifiers and bipolar transistors for specific applications.</li> </ul>		
<b>Personal Competence</b> <i>Social Competence</i>	<ul style="list-style-type: none"> <li>Students are able work efficiently in heterogeneous teams.</li> <li>Students working together in small groups can solve problems and answer professional questions.</li> </ul>		
<i>Autonomy</i>	<ul style="list-style-type: none"> <li>Students are able to assess their level of knowledge.</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>	General Engineering Science (German program, 7 semester): Specialisation Electrical Engineering: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Mechatronics: Compulsory Data Science: Core Qualification: Elective Compulsory Electrical Engineering: Core Qualification: Compulsory Engineering Science: Specialisation Electrical Engineering: Compulsory Engineering Science: Specialisation Mechatronics: Compulsory General Engineering Science (English program, 7 semester): Specialisation Electrical Engineering: Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechatronics: Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Mechanical Engineering: Specialisation Mechatronics: Compulsory Mechatronics: Specialisation Electrical Systems: Compulsory Mechatronics: Core Qualification: Compulsory Mechatronics: Specialisation Robot- and Machine-Systems: Elective Compulsory Technomathematics: Specialisation III. Engineering Science: Elective Compulsory		

Course L0763: Semiconductor Circuit Design	
<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 Kuhl
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Repetition Semiconductorphysics and Diodes</li> <li>• Functionality and characteristic curve of bipolar transistors</li> <li>• Basic circuits with bipolar transistors</li> <li>• Functionality and characteristic curve of MOS transistors</li> <li>• Basic circuits with MOS transistors for amplifiers</li> <li>• Operational amplifiers and their applications</li> <li>• Typical applications for analog and digital circuits</li> <li>• Realization of logical functions</li> <li>• Basic circuits with MOS transistors for combinational logic</li> <li>• Memory circuits</li> <li>• Basic circuits with MOS transistors for sequential logic</li> <li>• Basic concepts of analog-to-digital and digital-to-analog-converters</li> </ul>
<b>Literature</b>	<p>U. Tietze und Ch. Schenk, E. Gamm, Halbleiterschaltungstechnik, Springer Verlag, 14. Auflage, 2012, ISBN 3540428496</p> <p>R. J. Baker, CMOS - Circuit Design, Layout and Simulation, J. Wiley &amp; Sons Inc., 3. Auflage, 2011, ISBN: 0471700555</p> <p>H. Göbel, Einführung in die Halbleiter-Schaltungstechnik, Berlin, Heidelberg Springer-Verlag Berlin Heidelberg, 2011, ISBN: 9783642208874 ISBN: 9783642208867</p> <p>URL: <a href="http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10499499">http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10499499</a></p> <p>URL: <a href="http://dx.doi.org/10.1007/978-3-642-20887-4">http://dx.doi.org/10.1007/978-3-642-20887-4</a></p> <p>URL: <a href="http://ebooks.ciando.com/book/index.cfm/bok_id/319955">http://ebooks.ciando.com/book/index.cfm/bok_id/319955</a></p> <p>URL: <a href="http://www.ciando.com/img/bo">http://www.ciando.com/img/bo</a></p>

Course L0864: Semiconductor Circuit Design	
<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 Kuhl, Weitere Mitarbeiter
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Basic circuits and characteristic curves of bipolar transistors</li> <li>• Basic circuits and characteristic curves of MOS transistors for amplifiers</li> <li>• Realization and dimensioning of operational amplifiers</li> <li>• Realization of logic functions</li> <li>• Basic circuits with MOS transistors for combinational and sequential logic</li> <li>• Memory circuits</li> <li>• Circuits for analog-to-digital and digital-to-analog converters</li> <li>• Design of exemplary circuits</li> </ul>
<b>Literature</b>	<p>U. Tietze und Ch. Schenk, E. Gamm, Halbleiterschaltungstechnik, Springer Verlag, 14. Auflage, 2012, ISBN 3540428496</p> <p>R. J. Baker, CMOS - Circuit Design, Layout and Simulation, J. Wiley &amp; Sons Inc., 3. Auflage, 2011, ISBN: 0471700555</p> <p>H. Göbel, Einführung in die Halbleiter-Schaltungstechnik, Berlin, Heidelberg Springer-Verlag Berlin Heidelberg, 2011, ISBN: 9783642208874 ISBN: 9783642208867</p> <p>URL: <a href="http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10499499">http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10499499</a></p> <p>URL: <a href="http://dx.doi.org/10.1007/978-3-642-20887-4">http://dx.doi.org/10.1007/978-3-642-20887-4</a></p> <p>URL: <a href="http://ebooks.ciando.com/book/index.cfm/bok_id/319955">http://ebooks.ciando.com/book/index.cfm/bok_id/319955</a></p> <p>URL: <a href="http://www.ciando.com/img/bo">http://www.ciando.com/img/bo</a></p>

Module M1269: Lab Cyber-Physical Systems				
Courses				
Title	Typ		Hrs/wk	CP
Lab Cyber-Physical Systems (L1740)	Project-/problem-based Learning		4	6
<b>Module Responsible</b>	Prof. Heiko Falk			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Module "Embedded Systems"			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	<p><i>Knowledge</i> Cyber-Physical Systems (CPS) are tightly integrated with their surrounding environment, via sensors, A/D and D/A converters, and actors. Due to their particular application areas, highly specialized sensors, processors and actors are common. Accordingly, there is a large variety of different specification approaches for CPS - in contrast to classical software engineering approaches.</p> <p>Based on practical experiments using robot kits and computers, the basics of specification and modelling of CPS are taught. The lab introduces into the area (basic notions, characteristic properties) and their specification techniques (models of computation, hierarchical automata, data flow models, petri nets, imperative approaches). Since CPS frequently perform control tasks, the lab's experiments will base on simple control applications. The experiments will use state-of-the-art industrial specification tools (MATLAB/Simulink, LabVIEW, NXC) in order to model cyber-physical models that interact with the environment via sensors and actors.</p> <p><i>Skills</i> After successful attendance of the lab, students are able to develop simple CPS. They understand the interdependencies between a CPS and its surrounding processes which stem from the fact that a CPS interacts with the environment via sensors, A/D converters, digital processors, D/A converters and actors. The lab enables students to compare modelling approaches, to evaluate their advantages and limitations, and to decide which technique to use for a concrete task. They will be able to apply these techniques to practical problems. They obtain first experiences in hardware-related software development, in industry-relevant specification tools and in the area of simple control applications.</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 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>	Written elaboration			
<b>Examination duration and scale</b>	Execution and documentation of all lab experiments			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Mechatronics: Core Qualification: Elective Compulsory			

Course L1740: Lab Cyber-Physical Systems	
<b>Typ</b>	Project-/problem-based Learning
<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. Heiko Falk
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>Experiment 1: Programming in NXC</li> <li>Experiment 2: Programming the Robot in Matlab/Simulink</li> <li>Experiment 3: Programming the Robot in LabVIEW</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>Peter Marwedel. Embedded System Design - Embedded System Foundations of Cyber-Physical Systems. 2<sup>nd</sup> Edition, Springer, 2012.</li> <li>Begleitende Foliensätze</li> </ul>

Module M0854: Mathematics IV			
Courses			
Title	Typ	Hrs/wk	CP
Differential Equations 2 (Partial Differential Equations) (L1043)	Lecture	2	1
Differential Equations 2 (Partial Differential Equations) (L1044)	Recitation Section (small)	1	1
Differential Equations 2 (Partial Differential Equations) (L1045)	Recitation Section (large)	1	1
Complex Functions (L1038)	Lecture	2	1
Complex Functions (L1041)	Recitation Section (small)	1	1
Complex Functions (L1042)	Recitation Section (large)	1	1
<b>Module Responsible</b>	Prof. Marko Lindner		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Mathematics I - III		
<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 Mathematics IV. 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>Skills</i> <ul style="list-style-type: none"> <li>Students can model problems in Mathematics IV 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> <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 68, Study Time in Lecture 112		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	60 min (Complex Functions) + 60 min (Differential Equations 2)		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Electrical Engineering: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Mechatronics: Compulsory General Engineering Science (German program, 7 semester): Specialisation Naval Architecture: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Theoretical Mechanical Engineering: Elective Compulsory Electrical Engineering: Core Qualification: Compulsory General Engineering Science (English program, 7 semester): Specialisation Electrical Engineering: Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Mechanical Engineering: Specialisation Mechatronics: Compulsory Mechanical Engineering: Specialisation Theoretical Mechanical Engineering: Elective Compulsory Mechatronics: Core Qualification: Compulsory Naval Architecture: Core Qualification: Compulsory Theoretical Mechanical Engineering: Technical Complementary Course Core Studies: Elective Compulsory		

Course L1043: Differential Equations 2 (Partial Differential Equations)	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 2, Study Time in Lecture 28
<b>Lecturer</b>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<p>Main features of the theory and numerical treatment of partial differential equations</p> <ul style="list-style-type: none"> <li>• Examples of partial differential equations</li> <li>• First order quasilinear differential equations</li> <li>• Normal forms of second order differential equations</li> <li>• Harmonic functions and maximum principle</li> <li>• Maximum principle for the heat equation</li> <li>• Wave equation</li> <li>• Liouville's formula</li> <li>• Special functions</li> <li>• Difference methods</li> <li>• Finite elements</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• <a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a></li> </ul>

Course L1044: Differential Equations 2 (Partial Differential Equations)	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L1045: Differential Equations 2 (Partial Differential Equations)	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L1038: Complex Functions	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 2, Study Time in Lecture 28
<b>Lecturer</b>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<p>Main features of complex analysis</p> <ul style="list-style-type: none"> <li>• Functions of one complex variable</li> <li>• Complex differentiation</li> <li>• Conformal mappings</li> <li>• Complex integration</li> <li>• Cauchy's integral theorem</li> <li>• Cauchy's integral formula</li> <li>• Taylor and Laurent series expansion</li> <li>• Singularities and residuals</li> <li>• Integral transformations: Fourier and Laplace transformation</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• <a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a></li> </ul>

Course L1041: Complex Functions	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L1042: Complex Functions	
<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>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0610: Electrical Machines and Actuators				
Courses				
Title	Typ		Hrs/wk	CP
Electrical Machines and Actuators (L0293)	Lecture		3	4
Electrical Machines and Actuators (L0294)	Recitation Section (large)		2	2
<b>Module Responsible</b>	Prof. Thorsten Kern			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Basics of mathematics, in particular complexe numbers, integrals, differentials			
	Basics of electrical engineering and mechanical engineering			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	<p>Students can to draw and explain the basic principles of electric and magnetic fields.</p> <p>They can describe the function of the standard types of electric machines and present the corresponding equations and characteristic curves. For typically used drives they can explain the major parameters of the energy efficiency of the whole system from the power grid to the driven engine.</p> <p>Students are able to calculate two-dimensional electric and magnetic fields in particular ferromagnetic circuits with air gap. For this they apply the usual methods of the design auf electric machines.</p> <p>They can calculate the operational performance of electric machines from their given characteristic data and selected quantities and characteristic curves. They apply the usual equivalent circuits and graphical methods.</p>			
<i>Knowledge</i>				
<i>Skills</i>				
<b>Personal Competence</b>				
<i>Social Competence</i>				
<i>Autonomy</i>	<p>Students are able independently to calculate electric and magnatic fields for applications. They are able to analyse independently the operational performance of electric machines from the charactersitic data and theycan calculate thereof selected quantities and characteristic curves.</p>			
<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>	Subject theoretical and practical work			
<b>Examination duration and scale</b>	Design of four machines and actuators, review of design files			
<b>Assignment for the Following Curricula</b>	<p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Energy Systems: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Mechatronics: Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Theoretical Mechanical Engineering: Elective Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Electrical Engineering: Elective Compulsory</p> <p>Digital Mechanical Engineering: Core Qualification: Compulsory</p> <p>Electrical Engineering: Core Qualification: Elective Compulsory</p> <p>Engineering Science: Specialisation Electrical Engineering: Elective Compulsory</p> <p>Green Technologies: Energy, Water, Climate: Specialisation Energy Technology: Elective Compulsory</p> <p>Green Technologies: Energy, Water, Climate: Specialisation Maritime Technologies: Elective Compulsory</p> <p>Computer Science in Engineering: Specialisation II. Mathematics &amp; Engineering Science: Elective Compulsory</p> <p>Logistics and Mobility: Specialisation Traffic Planning and Systems: Elective Compulsory</p> <p>Logistics and Mobility: Specialisation Production Management and Processes: Elective Compulsory</p> <p>Mechanical Engineering: Core Qualification: Elective Compulsory</p> <p>Mechatronics: Specialisation Naval Engineering: Compulsory</p> <p>Mechatronics: Core Qualification: Compulsory</p> <p>Mechatronics: Specialisation Robot- and Machine-Systems: Compulsory</p> <p>Mechatronics: Specialisation Electrical Systems: Elective Compulsory</p> <p>Technomathematics: Specialisation III. Engineering Science: Elective Compulsory</p> <p>Engineering and Management - Major in Logistics and Mobility: Specialisation Traffic Planning and Systems: Elective Compulsory</p> <p>Engineering and Management - Major in Logistics and Mobility: Specialisation Information Technology: Elective Compulsory</p> <p>Engineering and Management - Major in Logistics and Mobility: Specialisation Production Management and Processes: Elective Compulsory</p> <p>Engineering and Management - Major in Logistics and Mobility: Specialisation Production Management and Processes: Elective Compulsory</p>			

Course L0293: Electrical Machines and Actuators	
<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. Thorsten Kern, Dennis Kähler
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<p>Electric field: Coulomb's law, flux (field) line, work, potential, capacitor, energy, force, capacitive actuators</p> <p>Magnetic field: force, flux line, Ampere's law, field at boundaries, flux, magnetic circuit, hysteresis, induction, self-induction, mutual inductance, transformer, electromagnetic actuators</p> <p>Synchronous machines, construction and layout, equivalent single line diagrams, no-load and short-circuit characteristics, vector diagrams, motor and generator operation, stepper motors</p> <p>DC-Machines: Construction and layout, torque generation mechanisms, torque vs speed characteristics, commutation,</p> <p>Asynchronous Machines. Magnetic field, construction and layout, equivalent single line diagram, complex stator current diagram (Heylands' diagram), torque vs. speed characteristics, rotor layout (squirrel-cage vs. sliprings),</p> <p>Drives with variable speed, inverter fed operation, special drives</p>
<b>Literature</b>	<p>Hermann Linse, Roland Fischer: "Elektrotechnik für Maschinenbauer", Vieweg-Verlag; Signatur der Bibliothek der TUHH: ETB 313</p> <p>Ralf Kories, Heinz Schmitt-Walter: "Taschenbuch der Elektrotechnik"; Verlag Harri Deutsch; Signatur der Bibliothek der TUHH: ETB 122</p> <p>"Grundlagen der Elektrotechnik" - anderer Autoren</p> <p>Fachbücher "Elektrische Maschinen"</p>

Course L0294: Electrical Machines and Actuators	
<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. Thorsten Kern, Dennis Kähler
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0567: Theoretical Electrical Engineering I: Time-Independent Fields			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Theoretical Electrical Engineering I: Time-Independent Fields (L0180)	Lecture	3	5
Theoretical Electrical Engineering I: Time-Independent Fields (L0181)	Recitation Section (small)	2	1
<b>Module Responsible</b>	Prof. Christian Schuster		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Basic principles of electrical engineering and advanced mathematics		
<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 fundamental formulas, relations, and methods of the theory of time-independent electromagnetic fields. They can explicate the principal behavior of electrostatic, magnetostatic, and current density fields with regard to respective sources. They can describe the properties of complex electromagnetic fields by means of superposition of solutions for simple fields. The students are aware of applications for the theory of time-independent electromagnetic fields and are able to explicate these.		
<i>Skills</i>	Students can apply Maxwell's Equations in integral notation in order to solve highly symmetrical, time-independent, electromagnetic field problems. Furthermore, they are capable of applying a variety of methods that require solving Maxwell's Equations for more general problems. The students can assess the principal effects of given time-independent sources of fields and analyze these quantitatively. They can deduce meaningful quantities for the characterization of electrostatic, magnetostatic, and electrical flow fields (capacitances, inductances, resistances, etc.) from given fields and dimension them for practical applications.		
<b>Personal Competence</b>			
<i>Social Competence</i>	Students are able to work together on subject related tasks in small groups. They are able to present their results effectively (e.g. during exercise sessions).		
<i>Autonomy</i>	Students are capable to gather necessary information from provided references and relate this information to the lecture. They are able to continually reflect their knowledge by means of activities that accompany the lecture, such as short oral quizzes during the lectures and exercises that are related to the exam. Based on respective feedback, students are expected to adjust their individual learning process. They are able to draw connections between their knowledge obtained in this lecture and the content of other lectures (e.g. Electrical Engineering I, Linear Algebra, and Analysis).		
<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>	Written exam		
<b>Examination duration and scale</b>	90-150 minutes		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Electrical Engineering: Compulsory Electrical Engineering: Core Qualification: Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Mechatronics: Specialisation Electrical Systems: Compulsory Technomathematics: Specialisation III. Engineering Science: Elective Compulsory		

Course L0180: Theoretical Electrical Engineering I: Time-Independent Fields	
<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>	Prof. Christian Schuster
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>- Maxwell's Equations in integral and differential notation</li> <li>- Boundary conditions</li> <li>- Laws of conservation for energy and charge</li> <li>- Classification of electromagnetic field properties</li> <li>- Integral characteristics of time-independent fields (R, L, C)</li> <li>- Generic approaches to solving Poisson's Equation</li> <li>- Electrostatic fields and specific methods of solving</li> <li>- Magnetostatic fields and specific methods of solving</li> <li>- Fields of electrical current density and specific methods of solving</li> <li>- Action of force within time-independent fields</li> <li>- Numerical methods for solving time-independent problems</li> </ul> <p>The practical application of numerical methods will be trained within specifically prepared lectures in an interactive manner using small MATLAB programs.</p>
<b>Literature</b>	<ul style="list-style-type: none"> <li>- G. Lehner, "Elektromagnetische Feldtheorie: Für Ingenieure und Physiker", Springer (2010)</li> <li>- H. Henke, "Elektromagnetische Felder: Theorie und Anwendung", Springer (2011)</li> <li>- W. Nolting, "Grundkurs Theoretische Physik 3: Elektrodynamik", Springer (2011)</li> <li>- D. Griffiths, "Introduction to Electrodynamics", Pearson (2012)</li> <li>- J. Edminister, "Schaum's Outline of Electromagnetics", McGraw-Hill (2013)</li> <li>- Richard Feynman, "Feynman Lectures on Physics: Volume 2", Basic Books (2011)</li> </ul>

Course L0181: Theoretical Electrical Engineering I: Time-Independent Fields	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 2, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Christian Schuster
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

### Specialization III. Subject Specific Focus

#### Module M1433: Technical Complementary Course for Computational Science and Engineering Bachelor

##### Courses

Title	Typ	Hrs/wk	CP
<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>	Depends on choice of courses		
<b>Credit points</b>	12		
<b>Assignment for the Following Curricula</b>	Computer Science in Engineering: Specialisation III. Subject Specific Focus: Elective Compulsory		

## Thesis

Module M-001: Bachelor Thesis				
Courses				
Title		Typ	Hrs/wk	CP
Module Responsible		Professoren der TUHH		
Admission Requirements		<ul style="list-style-type: none"><li>According to General Regulations §21 (1):  At least 126 ECTS credit points have to be achieved in study programme. The examinations board decides on exceptions.</li></ul>		
Recommended Previous Knowledge				
Educational Objectives		After taking part successfully, students have reached the following learning results		
Professional Competence		<ul style="list-style-type: none"><li>The students can select, outline and, if need be, critically discuss the most important scientific fundamentals of their course of study (facts, theories, and methods).</li><li>On the basis of their fundamental knowledge of their subject the students are capable in relation to a specific issue of opening up and establishing links with extended specialized expertise.</li><li>The students are able to outline the state of research on a selected issue in their subject area.</li></ul>		
Knowledge				
Skills				
Personal Competence		<ul style="list-style-type: none"><li>The students can make targeted use of the basic knowledge of their subject that they have acquired in their studies to solve subject-related problems.</li><li>With the aid of the methods they have learnt during their studies the students can analyze problems, make decisions on technical issues, and develop solutions.</li><li>The students can take up a critical position on the findings of their own research work from a specialized perspective.</li></ul>		
Social Competence				
Autonomy				
Workload in Hours		Independent Study Time 360, Study Time in Lecture 0		
Credit points		12		
Course achievement		None		
Examination		Thesis		
Examination duration and scale		According to General Regulations		
Assignment for the Following Curricula		General Engineering Science (German program): Thesis: Compulsory General Engineering Science (German program, 7 semester): Thesis: Compulsory Civil- and Environmental Engineering: Thesis: Compulsory Bioprocess Engineering: Thesis: Compulsory Chemical and Bioprocess Engineering: Thesis: Compulsory Computer Science: Thesis: Compulsory Data Science: Thesis: Compulsory Digital Mechanical Engineering: Thesis: Compulsory Electrical Engineering: Thesis: Compulsory Engineering Science: Thesis: Compulsory General Engineering Science (English program): Thesis: Compulsory General Engineering Science (English program, 7 semester): Thesis: Compulsory Green Technologies: Energy, Water, Climate: Thesis: Compulsory Computer Science in Engineering: Thesis: Compulsory Integrated Building Technology: Thesis: Compulsory Logistics and Mobility: Thesis: Compulsory Mechanical Engineering: Thesis: Compulsory Mechatronics: Thesis: Compulsory Naval Architecture: Thesis: Compulsory Technomathematics: Thesis: Compulsory Teilstudiengang Lehramt Metalltechnik: Thesis: Compulsory Process Engineering: Thesis: Compulsory Engineering and Management - Major in Logistics and Mobility: Thesis: Compulsory		