Module Manual

Bachelor of Science (B.Sc.)

Computer Science

Cohort: Winter Term 2021

Updated: 18th September 2021

Table of Contents

Table of Contents 2
Program description 3
Core Qualification 4
Module M0561: Discrete Algebraic Structures 4
Module M0731: Functional Programming 5
Module M0577: Non-technical Courses for Bachelors 7
Module M1436: Procedural Programming for Computer Engineers 9
Module M1728: Mathematics I (EN) 10
Module M1729: Mathematics II (EN) 12
Module M0624: Automata Theory and Formal Languages 14
Module M0829: Foundations of Management 16
Module M1432: Programming Paradigms 19
Module M1732: Mathematics III (EN) 21
Module M0834: Computernetworks and Internet Security 23
Module M0730: Computer Engineering 25
Module M1423: Algorithms and Data Structures 27
Module M0727: Stochastics 29
Module M0732: Software Engineering 31
Module M0852: Graph Theory and Optimization 33
Module M0562: Computability and Complexity Theory 35
Module M0873: Software Industrial Internship 36
Module M1578: Seminars Computer Science 37
Module M1620: Ethics in Information Technology 38
Specialization I. Computer and Software Engineering 39
Module M0625: Databases 39
Module M0971: Operating Systems 41
Module M1586: Scientific Programming 42
Module M0791: Computer Architecture 44
Module M0953: Introduction to Information Security 46
Module M0803: Embedded Systems 48
Module M0754: Compiler Construction 50
Module M1300: Software Development 52
Specialization II. Mathematics and Engineering Science 54
Module M0662: Numerical Mathematics I 54
Module M1730: Mathematics IV (EN) 56
Module M0651: Computational Geometry 58
Module M0941: Combinatorial Structures and Algorithms 61
Module M1242: Quantum Mechanics for Engineers 63
Module M1592: Statistics 65
Module M0634: Introduction into Medical Technology and Systems 67
Module M0668: Algebra and Control 69
Module M1269: Lab Cyber-Physical Systems 71
Module M0715: Solvers for Sparse Linear Systems 72
Module M0672: Signals and Systems 74
Specialization III. Subject Specific Focus 77
Module M1562: Technical Complementary Course I for CSBS 77
Module M1568: Technical Complementary Course II for CSBS 78
Thesis 79
Module M-001: Bachelor Thesis 79

Program description

Content

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

Module Responsible	Prof. Karl-Heinz Zimmermann
Admission Requirements	None
Recommended Previous Knowledge	Mathematics from High School.
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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.
Skills	Students are able to formalize and analyze basic discrete algebraic structures.
Personal Competence	
Social Competence	Students are able to solve specific problems alone or in a group and to present the results accordingly.
Autonomy	Students are able to acquire new knowledge from specific standard books and to associate the acquired knowledge to other classes.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	120 min
Assignment for the Following Curricula	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

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Karl-Heinz Zimmermann
Language	DE/EN
Cycle	WiSe
Content	
Literature	

Course L0165: Discrete Algebraic Structures

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

Module M0731: Functional Programming

Courses

Title	Typ	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

Module Responsible	Prof. Sibylle Schupp
Admission Requirements	None
Recommended Previous Knowledge	Discrete mathematics at high-school level
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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.
Skills	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.
Personal Competence	
Social Competence	Students practice peer programming with varying peers. They explain problems and solutions to their peer. They defend their programs orally. They communicate in English.
Autonomy	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.
Workload in Hours	Independent Study Time 96, Study Time in Lecture 84
Credit points	6
Course achievement	**Compulsory** **Bonus** **Form** **Description** Yes 15 % Excercises
Examination	Written exam
Examination duration and scale	90 min
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: Elective Compulsory Engineering Science: Specialisation Mechatronics: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Computer Science: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechatronics: Elective Compulsory Computational Science and Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory

Course L0624: Functional Programming

Typ	Lecture
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	WiSe
Content	• Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions • Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions • Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type • Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps) • Modules • Interactive Programming • Lazy Evaluation, Call-by-Value, Strictness • Design Recipes • Testing (axiom-based, invariant-based, against reference implementation) • Reasoning about Programs (equation-based, inductive) • Idioms of Functional Programming • Haskell Syntax and Semantics
Literature	Graham Hutton, Programming in Haskell, Cambridge University Press 2007.

Course L0625: Functional Programming	
Typ	Recitation Section (large)
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	WiSe
Content	• Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions • Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions • Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type • Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps) • Modules • Interactive Programming • Lazy Evaluation, Call-by-Value, Strictness • Design Recipes • Testing (axiom-based, invariant-based, against reference implementation) • Reasoning about Programs (equation-based, inductive) • Idioms of Functional Programming • Haskell Syntax and Semantics
Literature	Graham Hutton, Programming in Haskell, Cambridge University Press 2007.

Course L0626: Functional Programming	
Typ	Recitation Section (small)
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	WiSe
Content	• Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions • Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions • Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type • Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps) • Modules • Interactive Programming • Lazy Evaluation, Call-by-Value, Strictness • Design Recipes • Testing (axiom-based, invariant-based, against reference implementation) • Reasoning about Programs (equation-based, inductive) • Idioms of Functional Programming • Haskell Syntax and Semantics
Literature	Graham Hutton, Programming in Haskell, Cambridge University Press 2007.

Module M0577: Non-technical Courses for Bachelors

Module Responsible	Dagmar Richter
Admission Requirements	None
Recommended Previous Knowledge	None
Educational Objectives	After taking part successfully, students have reached the following learning results

Professional Competence

Knowledge

The Non-technical Academic Programms (NTA)

imparts skills that, in view of the TUHH’s training profile, professional engineering studies require but are not able to cover fully. Self-reliance, self-management, collaboration and professional and personnel management competences. The department implements these training objectives in its **teaching architecture**, in its **teaching and learning arrangements**, in **teaching areas** and by means of teaching offerings in which students can qualify by opting for **specific competences** and a **competence level** at the Bachelor’s or Master’s level. The teaching offerings are pooled in two different catalogues for nontechnical complementary courses.

The Learning Architecture

consists of a cross-disciplinarily study offering. The centrally designed teaching offering ensures that courses in the nontechnical academic programms follow the specific profiling of TUHH degree courses.

The learning architecture demands and trains independent educational planning as regards the individual development of competences. It also provides orientation knowledge in the form of “profiles”

The subjects that can be studied in parallel throughout the student’s entire study program - if need be, it can be studied in one to two semesters. In view of the adaptation problems that individuals commonly face in their first semesters after making the transition from school to university and in order to encourage individually planned semesters abroad, there is no obligation to study these subjects in one or two specific semesters during the course of studies.

Teaching and Learning Arrangements

provide for students, separated into B.Sc. and M.Sc., to learn with and from each other across semesters. The challenge of dealing with interdisciplinarity and a variety of stages of learning in courses are part of the learning architecture and are deliberately encouraged in specific courses.

Fields of Teaching

are based on research findings from the academic disciplines cultural studies, social studies, arts, historical studies, 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.

The fields of teaching are augmented by soft skills offers and a foreign language offer. Here, the focus is on encouraging goal-oriented communication skills, e.g. the skills required by outgoing engineers in international and intercultural situations.

The Competence Level

of the courses offered in this area is different as regards the basic training objective in the Bachelor’s and Master’s fields. These differences are reflected in the practical examples used, in content topics that refer to different professional application contexts, and in the higher scientific and theoretical level of abstraction in the B.Sc.

This is also reflected in the different quality of soft skills, which relate to the different team positions and different group leadership functions of Bachelor’s and Master’s graduates in their future working life.

Specialized Competence (Knowledge)

Students can

- locate selected specialized areas with the relevant non-technical mother discipline,
- outline basic theories, categories, terminology, models, concepts or artistic techniques in the disciplines represented in the learning area,
- different specialist disciplines relate to their own discipline and differentiate it as well as make connections,
- sketch the basic outlines of how scientific disciplines, paradigms, models, instruments, methods and forms of representation in the specialized sciences are subject to individual and socio-cultural interpretation and historicity,
- Can communicate in a foreign language in a manner appropriate to the subject.

Skills

Professional Competence (Skills)

In selected sub-areas students can

- apply basic methods of the said scientific disciplines,
- auestion a specific technical phenomena, models, theories from the viewpoint of another, aforementioned specialist discipline,
- to handle simple questions in aforementioned scientific disciplines in a sucsessful manner,
- justify their decisions on forms of organization and application in practical questions in contexts that go beyond the technical relationship to the subject.

Personal Competence

Social Competence

Personal Competences (Social Skills)

Students will be able

- to learn to collaborate in different manner,

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

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

Module M1436: Procedural Programming for Computer Engineers

Courses

Title	Typ	Hrs/wk	CP
Procedural Programming for Computer Engineers (L2163)	Lecture	1	2
Procedular Programming for Computer Engineers (L2164)	Recitation Section (large)	1	1
Procedural Programming for Computer Engineers (L2165)	Practical Course	2	3

Module Responsible	NN
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	120 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 L2163: Procedural Programming for Computer Engineers

Typ	Lecture
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Siegfried Rump
Language	DE/EN
Cycle	WiSe
Content	
Literature	

Course L2164: Procedular Programming for Computer Engineers

Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Dozenten des SD E
Language	DE/EN
Cycle	WiSe
Content	
Literature	

Course L2165: Procedural Programming for Computer Engineers

Typ	Practical Course
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Dozenten des SD E
Language	DE/EN
Cycle	WiSe
Content	
Literature	

Module M1728: Mathematics I (EN)

Courses

Title	Typ	Hrs/wk	CP
Analysis I (EN) (L2771)	Lecture	2	2
Analysis I (EN) (L2772)	Recitation Section (large)	1	1
Analysis I (EN) (L2773)	Recitation Section (small)	1	1
Linear Algebra I (EN) (L2774)	Lecture	2	2
Linear Algebra I (EN) (L2775)	Recitation Section (large)	1	1
Linear Algebra I (EN) (L2776)	Recitation Section (small)	1	1

Module Responsible	Prof. Daniel Ruprecht
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 128, Study Time in Lecture 112
Credit points	8
Course achievement	None
Examination	Written exam
Examination duration and scale	120 min
Assignment for the Following Curricula	Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory

Course L2771: Analysis I (EN)

Typ	Lecture
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Daniel Ruprecht
Language	EN
Cycle	WiSe
Content	
Literature	

Course L2772: Analysis I (EN)

Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Daniel Ruprecht
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Course L2773: Analysis I (EN)

Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Daniel Ruprecht
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Course L2774: Linear Algebra I (EN)	
Typ	Lecture
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Daniel Ruprecht
Language	EN
Cycle	WiSe
Content	
Literature	

Course L2775: Linear Algebra I (EN)	
Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Daniel Ruprecht
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Course L2776: Linear Algebra I (EN)	
Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Daniel Ruprecht
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1729: Mathematics II (EN)

Courses

Title	Typ	Hrs/wk	CP
Analysis II (English) (L2777)	Lecture	2	2
Analysis II (English) (L2778)	Recitation Section (large)	1	1
Analysis II (English) (L2779)	Recitation Section (small)	1	1
Linear Algebra II (English) (L2780)	Lecture	2	2
Linear Algebra II (English) (L2781)	Recitation Section (large)	1	1
Linear Algebra II (English) (L2782)	Recitation Section (small)	1	1

Module Responsible	Prof. Daniel Ruprecht
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 128, Study Time in Lecture 112
Credit points	8
Course achievement	None
Examination	Written exam
Examination duration and scale	120 min
Assignment for the Following Curricula	Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory

Course L2777: Analysis II (English)

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

Course L2778: Analysis II (English)

Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Daniel Ruprecht
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Course L2779: Analysis II (English)

Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Daniel Ruprecht
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Course L2780: Linear Algebra II (English)	
Typ	Lecture
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Daniel Ruprecht
Language	EN
Cycle	SoSe
Content	
Literature	

Course L2781: Linear Algebra II (English)	
Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Daniel Ruprecht
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Course L2782: Linear Algebra II (English)	
Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Daniel Ruprecht
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M0624: Automata Theory and Formal Languages

Courses

Title	Typ	Hrs/wk	CP
Automata Theory and Formal Languages (L0332)	Lecture	2	4
Automata Theory and Formal Languages (L0507)	Recitation Section (small)	2	2

Module Responsible	Prof. Tobias Knopp
Admission Requirements	None
Recommended Previous Knowledge	Participating students should be able to - specify algorithms for simple data structures (such as, e.g., arrays) to solve computational problems - apply propositional logic and predicate logic for specifying and understanding mathematical proofs - apply the knowledge and skills taught in the module Discrete Algebraic Structures
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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.
Skills	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.
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	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

Course L0332: Automata Theory and Formal Languages	
Typ	Lecture
Hrs/wk	2
CP	4
Workload in Hours	Independent Study Time 92, Study Time in Lecture 28
Lecturer	Prof. Tobias Knopp
Language	EN
Cycle	SoSe
Content	1. Propositional logic, Boolean algebra, propositional resolution, SAT-2KNF 2. Predicate logic, unification, predicate logic resolution 3. Temporal Logics (LTL, CTL) 4. Deterministic finite automata, definition and construction 5. Regular languages, closure properties, word problem, string matching 6. Nondeterministic automata: Rabin-Scott transformation of nondeterministic into deterministic automata 7. Epsilon automata, minimization of automata, elimination of e-edges, uniqueness of the minimal automaton (modulo renaming of states) 8. Myhill-Nerode Theorem: Correctness of the minimization procedure, equivalence classes of strings induced by automata 9. Pumping Lemma for regular languages: 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 10. Regular expressions vs. finite automata: Equivalence of formalisms, systematic transformation of representations, reductions 11. Pushdown automata and context-free grammars: 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) 12. Chomsky normal form 13. CYK algorithm for deciding the word problem for context-free grammrs 14. Deterministic pushdown automata 15. Deterministic vs. nondeterministic pushdown automata: Application for parsing, LL(k) or LR(k) grammars and parsers vs. deterministic pushdown automata, compiler compiler 16. Regular grammars 17. Outlook: Turing machines and linear bounded automata vs general and context-sensitive grammars 18. Chomsky hierarchy 19. Mealy- and Moore automata: Automata with output (w/o accepting states), infinite state sequences, automata networks 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) 21. LTL safety conditions and model checking with Büchi automata, relationships between automata and logic 22. Fixed points, propositional mu-calculus 23. Characterization of regular languages by monadic second-order logic (MSO)
Literature	1. Logik für Informatiker Uwe Schöning, Spektrum, 5. Aufl. 2. Logik für Informatiker Martin Kreuzer, Stefan Kühling, Pearson Studium, 2006 3. Grundkurs Theoretische Informatik, Gottfried Vossen, Kurt-Ulrich Witt, Vieweg-Verlag, 2010. 4. Principles of Model Checking, Christel Baier, Joost-Pieter Katoen, The MIT Press, 2007

Course L0507: Automata Theory and Formal Languages	
Typ	Recitation Section (small)
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Tobias Knopp
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	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

Module Responsible	Prof. Christoph Ihl
Admission Requirements	None
Recommended Previous Knowledge	Basic Knowledge of Mathematics and Business
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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 • explain the differences between Economics and Management and the sub-disciplines in Management and to name important definitions from the field of Management • explain the most important aspects of and goals in Management and name the most important aspects of entreprneurial projects • describe and explain basic business functions as production, procurement and sourcing, supply chain management, organization and human ressource management, information management, innovation management and marketing • 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 • state basics from accounting and costing and selected controlling methods.
Skills	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 • analyse Management goals and structure them appropriately • analyse organisational and staff structures of companies • apply methods for decision making under multiple objectives, under uncertainty and under risk • analyse production and procurement systems and Business information systems • analyse and apply basic methods of marketing • select and apply basic methods from mathematical finance to predefined problems • apply basic methods from accounting, costing and controlling to predefined problems
Personal Competence	
Social Competence	Students are able to • work successfully in a team of students • to apply their knowledge from the lecture to an entrepreneurship project and write a coherent report on the project • to communicate appropriately and • to cooperate respectfully with their fellow students.
Autonomy	Students are able to • work in a team and to organize the team themselves • to write a report on their project.
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	None
Examination	Subject theoretical and practical work
Examination duration and scale	several written exams during the semester
Assignment for the Following Curricula	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 Enviromental 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

	General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Materials in Engineering Sciences: Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Mechatronics: Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Product Development and Production: Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Theoretical Mechanical Engineering: Compulsory General Engineering Science (English program, 7 semester): Specialisation Naval Architecture: Compulsory General Engineering Science (English program, 7 semester): Specialisation Process Engineering: Compulsory General Engineering Science (English program, 7 semester): Specialisation Biomedical Engineering: 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 Orientation Studies: Core Qualification: Elective Compulsory Naval Architecture: Core Qualification: Compulsory Technomathematics: Core Qualification: Compulsory Process Engineering: Core Qualification: Compulsory Engineering and Management - Major in Logistics and Mobility: Core Qualification: Compulsory

Course L0882: Management Tutorial

Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Christoph Ihl, Katharina Roedelius
Language	DE
Cycle	WiSe/SoSe
Content	In the management tutorial, the contents of the lecture will be deepened by practical examples and the application of the discussed tools. 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 se 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 busin knowledge from the lecture should come to practical use. The group projects are guided by a mentor.
Literature	Relevante Literatur aus der korrespondierenden Vorlesung.

Course L0880: Introduction to Management	
Typ	Lecture
Hrs/wk	3
CP	3
Workload in Hours	Independent Study Time 48, Study Time in Lecture 42
Lecturer	Prof. Christoph Ihl, Prof. Thorsten Blecker, Prof. Christian Lüthje, Prof. Christian Ringle, Prof. Kathrin Fischer, Prof. Cornelius Herstatt, Prof. Wolfgang Kersten, Prof. Matthias Meyer, Prof. Thomas Wrona
Language	DE
Cycle	WiSe/SoSe
Content	• Introduction to Business and Management, Business versus Economics, relevant areas in Business and Management • Important definitions from Management, • Developing Objectives for Business, and their relation to important Business functions • Business Functions: Functions of the Value Chain, e.g. Production and Procurement, Supply Chain Management, Innovation Management, Marketing and Sales Cross-sectional Functions, e.g. Organisation, Human Ressource Management, Supply Chain Management, Information Management • Definitions as information, information systems, aspects of data security and strategic information systems • Definition and Relevance of innovations, e.g. innovation opporunities, risks etc. • Relevance of marketing, B2B vs. B2C-Marketing • different techniques from the field of marketing (e.g. scenario technique), pricing strategies • important organizational structures • basics of human ressource management • Introduction to Business Planning and the steps of a planning process • Decision Analysis: Elements of decision problems and methods for solving decision problems • Selected Planning Tasks, e.g. Investment and Financial Decisions • Introduction to Accounting: Accounting, Balance-Sheets, Costing • Relevance of Controlling and selected Controlling methods • Important aspects of Entrepreneurship projects
Literature	Bamberg, G., Coenenberg, A.: Betriebswirtschaftliche Entscheidungslehre, 14. Aufl., München 2008 Eisenführ, F., Weber, M.: Rationales Entscheiden, 4. Aufl., Berlin et al. 2003 Heinhold, M.: Buchführung in Fallbeispielen, 10. Aufl., Stuttgart 2006. Kruschwitz, L.: Finanzmathematik. 3. Auflage, München 2001. Pellens, B., Fülbier, R. U., Gassen, J., Sellhorn, T.: Internationale Rechnungslegung, 7. Aufl., Stuttgart 2008. Schweitzer, M.: Planung und Steuerung, in: Bea/Friedl/Schweitzer: Allgemeine Betriebswirtschaftslehre, Bd. 2: Führung, 9. Aufl., Stuttgart 2005. Weber, J., Schäffer, U. : Einführung in das Controlling, 12. Auflage, Stuttgart 2008. Weber, J./Weißenberger, B.: Einführung in das Rechnungswesen, 7. Auflage, Stuttgart 2006.

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	NN
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	
Knowledge	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.
Skills	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.
Personal Competence	
Social Competence	Students can work in teams and communicate in forums.
Autonomy	In a programming internship, students learn object-oriented programming under supervision. In exercises they develop individual and independent solutions and receive feedback.
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

Typ	Lecture
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Dozenten des SD E
Language	DE/EN
Cycle	SoSe
Content	• fundamentals behind object orientated programming • classes and objects • inheritance (single, multiple) • interfaces • information hiding • exception handling • generic programming and the implementation in the compiler • excursus in programming with dynamically typed programming languages
Literature	Skript

Course L2170: Programming Paradigms	
Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Dozenten des SD E
Language	DE/EN
Cycle	SoSe
Content	• fundamentals behind object orientated programming • classes and objects • inheritance (single, multiple) • interfaces • information hiding • exception handling • generic programming and the implementation in the compiler • excursus in programming with dynamically typed programming languages
Literature	Skript

Course L2171: Programming Paradigms	
Typ	Practical Course
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Dozenten des SD E
Language	DE/EN
Cycle	SoSe
Content	• fundamentals behind object orientated programming • classes and objects • inheritance (single, multiple) • interfaces • information hiding • exception handling • generic programming and the implementation in the compiler • excursus in programming with dynamically typed programming languages
Literature	Skript

Module M1732: Mathematics III (EN)

Courses

Title	Typ	Hrs/wk	CP
Analysis III (English) (L2790)	Lecture	2	2
Analysis III (English) (L2791)	Recitation Section (large)	1	1
Analysis III (English) (L2792)	Recitation Section (small)	1	1
Differential Equations 1 (Ordinary Differential Equations) (L2793)	Lecture	2	2
Differential Equations 1 (Ordinary Differential Equations) (L2794)	Recitation Section (large)	1	1
Differential Equations 1 (Ordinary Differential Equations) (L2795)	Recitation Section (small)	1	1

Module Responsible	Prof. Anusch Taraz
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
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	120 min
Assignment for the Following Curricula	Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory

Course L2790: Analysis III (English)

Typ	Lecture
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Dozenten des Fachbereiches Mathematik der UHH
Language	EN
Cycle	WiSe
Content	
Literature	

Course L2791: Analysis III (English)

Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Dozenten des Fachbereiches Mathematik der UHH
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Course L2792: Analysis III (English)

Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Dozenten des Fachbereiches Mathematik der UHH
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Course L2793: Differential Equations 1 (Ordinary Differential Equations)	
Typ	Lecture
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Dozenten des Fachbereiches Mathematik der UHH
Language	EN
Cycle	WiSe
Content	
Literature	

Course L2794: Differential Equations 1 (Ordinary Differential Equations)	
Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Dozenten des Fachbereiches Mathematik der UHH
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Course L2795: Differential Equations 1 (Ordinary Differential Equations)	
Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Dozenten des Fachbereiches Mathematik der UHH
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M0834: Computernetworks and Internet Security

Courses

Title	Typ	Hrs/wk	CP
Computer Networks and Internet Security (L1098)	Lecture	3	5
Computer Networks and Internet Security (L1099)	Recitation Section (small)	1	1

Module Responsible	Prof. Andreas Timm-Giel
Admission Requirements	None
Recommended Previous Knowledge	Basics of Computer Science
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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.
Skills	Students are able to analyse common Internet protocols and evaluate the use of them in different domains.
Personal Competence	
Social Competence	
Autonomy	Students can select relevant parts out of high amount of professional knowledge and can independently learn and understand it.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	120 min
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: Elective Compulsory Electrical Engineering: Core Qualification: Elective Compulsory Engineering Science: Specialisation Mechatronics: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Computer Science: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechatronics: Elective Compulsory Computational Science and Engineering: Core Qualification: Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory

Course L1098: Computer Networks and Internet Security

Typ	Lecture
Hrs/wk	3
CP	5
Workload in Hours	Independent Study Time 108, Study Time in Lecture 42
Lecturer	Prof. Andreas Timm-Giel, Prof. Dieter Gollmann, Dr.-Ing. Koojana Kuladinithi
Language	EN
Cycle	WiSe
Content	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. In the second part of the lecture an introduction to Internet security is given. This class comprises: • Application layer protocols (HTTP, FTP, DNS) • Transport layer protocols (TCP, UDP) • Network Layer (Internet Protocol, routing in the Internet) • Data link layer with media access at the example of Ethernet • Multimedia applications in the Internet • Network management • Internet security: IPSec • Internet security: Firewalls
Literature	• Kurose, Ross, Computer Networking - A Top-Down Approach, 6th Edition, Addison-Wesley • Kurose, Ross, Computernetzwerke - Der Top-Down-Ansatz, Pearson Studium; Auflage: 6. Auflage • W. Stallings: Cryptography and Network Security: Principles and Practice, 6th edition Further literature is announced at the beginning of the lecture.

Course L1099: Computer Networks and Internet Security	
Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Andreas Timm-Giel, Prof. Dieter Gollmann
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	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

Module Responsible	Prof. Heiko Falk
Admission Requirements	None
Recommended Previous Knowledge	Basic knowledge in electrical engineering
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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: • Introduction • Combinational logic: Gates, Boolean algebra, Boolean functions, hardware synthesis, combinational networks • Sequential logic: Flip-flops, automata, systematic hardware design • Technological foundations • Computer arithmetic: Integer addition, subtraction, multiplication and division • Basics of computer architecture: Programming models, MIPS single-cycle architecture, pipelining • Memories: Memory hierarchies, SRAM, DRAM, caches • Input/output: I/O from the perspective of the CPU, principles of passing data, point-to-point connections, busses
Skills	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. 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.
Personal Competence	
Social Competence	Students are able to solve similar problems alone or in a group and to present the results accordingly.
Autonomy	Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory** Yes · **Bonus** 10 % · **Form** Excercises · **Description**
Examination	Written exam
Examination duration and scale	90 minutes, contents of course and labs
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory General Engineering Science (German program, 7 semester): Specialisation Civil Engineering: Compulsory General Engineering Science (German program, 7 semester): Specialisation Process Engineering: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Mechatronics: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Aircraft Systems Engineering: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Theoretical Mechanical Engineering: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Materials in Engineering Sciences: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Product Development and Production: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Energy Systems: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Biomechanics: Compulsory General Engineering Science (German program, 7 semester): Specialisation Naval Architecture: Compulsory General Engineering Science (German program, 7 semester): Specialisation Biomedical Engineering: Compulsory General Engineering Science (German program, 7 semester): Specialisation Energy and Enviromental Engineering: Compulsory General Engineering Science (German program, 7 semester): Specialisation Bioprocess Engineering: Compulsory General Engineering Science (German program, 7 semester): Specialisation Electrical Engineering: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Elective Compulsory Electrical Engineering: Core Qualification: Compulsory General Engineering Science (English program, 7 semester): Specialisation Civil Engineering: 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 Aircraft Systems Engineering: Compulsory

	General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Materials in Engineering Sciences: Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Mechatronics: Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Product Development and Production: Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Theoretical Mechanical Engineering: Compulsory Computational Science and Engineering: Core Qualification: Compulsory Mechatronics: Core Qualification: Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory

Course L0321: Computer Engineering	
Typ	Lecture
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Heiko Falk
Language	DE/EN
Cycle	WiSe
Content	• Introduction • Combinational Logic • Sequential Logic • Technological Foundations • Representations of Numbers, Computer Arithmetics • Foundations of Computer Architecture • Memories • Input/Output
Literature	• A. Clements. The Principles of Computer Hardware. 3. Auflage, Oxford University Press, 2000. • A. Tanenbaum, J. Goodman. Computerarchitektur. Pearson, 2001. • D. Patterson, J. Hennessy. Rechnerorganisation und -entwurf. Elsevier, 2005.

Course L0324: Computer Engineering	
Typ	Recitation Section (small)
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Heiko Falk
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	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

Module Responsible	Prof. Matthias Mnich
Admission Requirements	None
Recommended Previous Knowledge	• Discrete Algebraic Structures • Mathematics I • Mathematics II • Procedual Programming • Objectoriented Programming
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students can name the basic concepts in algorithm design, algorithm analysis and problem reductions. They are able to explain them using appropriate examples. • Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples. • They know proof strategies and can reproduce them.
Skills	• Students can model 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. • Students are able to discover and verify further logical connections between the concepts studied in the course. • For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.
Personal Competence	
Social Competence	• Students are able to work together in teams. They are capable to use mathematics as a common language. • In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.
Autonomy	• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. • Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	60 min
Assignment for the Following Curricula	Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Computational Science and 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	
Typ	Lecture
Hrs/wk	4
CP	4
Workload in Hours	Independent Study Time 64, Study Time in Lecture 56
Lecturer	Prof. Matthias Mnich
Language	DE/EN
Cycle	WiSe
Content	• Insertion sort • Register machines • Asymptotic analysis, Landau notation • Polynomial-time algorithms and NP-completeness • Divide-and-conquer, merge sort • Strassen algorithm • Greedy algorithm • Dynamic programming • Quick sort • AVL-trees, B-trees • Hashing • Depth first search, breadth first search • Shortest paths • Flow problems, Ford-Fulkerson algorithm
Literature	• T. Cormen, Ch. Leiserson, R. Rivest, C. Stein: Introduction to Algorithms. MIT Press, 2013 • S. Skiena: The Algorithm Design Manual. Springer, 2008 • J. M. Kleinberg and É. Tardos. Algorithm Design. Addison-Wesley, 2005.

Course L2047: Algorithms and Data Structures	
Typ	Recitation Section (small)
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Matthias Mnich
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M0727: Stochastics

Courses

Title	Typ	Hrs/wk	CP
Stochastics (L0777)	Lecture	2	4
Stochastics (L0778)	Recitation Section (small)	2	2

Module Responsible	Prof. Matthias Schulte
Admission Requirements	None
Recommended Previous Knowledge	• Calculus • Discrete algebraic structures (combinatorics) • Propositional logic
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students can name the basic concepts in Stochastics. They are able to explain them using appropriate examples. • Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples. • They know proof strategies and can reproduce them.
Skills	• Students can model problems from stochastics with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods. • Students are able to discover and verify further logical connections between the concepts studied in the course. • For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.
Personal Competence	
Social Competence	• Students are able to work together (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). • In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.
Autonomy	• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. • Students can put their knowledge in relation to the contents of other lectures. • Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	120 min
Assignment for the Following Curricula	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 Logistics and Mobility: Specialisation Engineering Science: Elective Compulsory Logistics and Mobility: Specialisation Information Technology: 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	
Typ	Lecture
Hrs/wk	2
CP	4
Workload in Hours	Independent Study Time 92, Study Time in Lecture 28
Lecturer	Prof. Matthias Schulte
Language	DE/EN
Cycle	SoSe
Content	• Definitions of probability, conditional probability • Random variables, dependencies, independence assumptions, • Marginal and joint probabilities • Distributions and density functions • Characteristics: expected values, variance, standard deviation, moments • Multivariate distributions • Law of large numbers and central limit theorem • Basic notions of stochastic processes • Basic concepts of statistics (point estimators, confidence intervals, hypothesis testing)
Literature	1. Methoden der statistischen Inferenz, Likelihood und Bayes, Held, L., Spektrum 2008 2. Stochastik für Informatiker, Dümbgen, L., Springer 2003 3. Statistik: Der Weg zur Datenanalyse, Fahrmeir, L., Künstler R., Pigeot, I, Tutz, G., Springer 2010 4. Stochastik, Georgii, H.-O., deGruyter, 2009 5. Probability and Random Processes, Grimmett, G., Stirzaker, D., Oxford University Press, 2001 6. Programmieren mit R, Ligges, U., Springer 2008

Course L0778: Stochastics	
Typ	Recitation Section (small)
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Matthias Schulte
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	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

Module Responsible	Prof. Sibylle Schupp
Admission Requirements	None
Recommended Previous Knowledge	• Automata theory and formal languages • Procedural programming or Functional programming • Object-oriented programming, algorithms, and data structures
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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.
Skills	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.
Personal Competence	
Social Competence	Students practice peer programming. They explain problems and solutions to their peer. They communicate in English.
Autonomy	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.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory** Yes · **Bonus** 15 % · **Form** Excercises · **Description**
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Core Qualification: Compulsory General Engineering Science (English program, 7 semester): Specialisation Computer Science: Elective Compulsory Computational Science and Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory

Course L0627: Software Engineering

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	SoSe
Content	• Software Life Cycle Models (Waterfall, V-Model, Evolutionary Models, IncrementalModels, Iterative Models, Agile Processes) • Requirements (Elicitation Techniques, UML Use Case Diagrams, Functional and Non-Functional Requirements) • Specification (Finite State Machines, Extended FSMs, Petri Nets, Behavioral UML Diagrams, Data Modeling) • Design (Design Concepts, Modules, (Agile) Design Principles) • Object-Oriented Analysis and Design (Object Identification, UML Interaction Diagrams, UML Class Diagrams, Architectural Patterns) • Testing (Blackbox Testing, Whitebox Testing, Control-Flow Testing, Data-Flow Testing, Testing in the Large) • Maintenance and Evolution (Regression Testing, Reverse Engineering, Reengineering) • Project Management (Blackbox Estimation Techniques, Whitebox Estimation Techniques, Project Plans, Gantt Charts, PERT Charts)
Literature	Kassem A. Saleh, Software Engineering, J. Ross Publishing 2009.

Course L0628: Software Engineering	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M0852: Graph Theory and Optimization

Courses

Title	Typ	Hrs/wk	CP
Graph Theory and Optimization (L1046)	Lecture	2	3
Graph Theory and Optimization (L1047)	Recitation Section (small)	2	3

Module Responsible	Prof. Anusch Taraz
Admission Requirements	None
Recommended Previous Knowledge	• Discrete Algebraic Structures • Mathematics I
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students can name the basic concepts in Graph Theory and Optimization. They are able to explain them using appropriate examples. • Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples. • They know proof strategies and can reproduce them.
Skills	• Students can model problems in Graph Theory and Optimization with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods. • Students are able to discover and verify further logical connections between the concepts studied in the course. • For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.
Personal Competence	
Social Competence	• Students are able to work together in teams. They are capable to use mathematics as a common language. • In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.
Autonomy	• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. • Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	120 min
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Logistics and Mobility: Specialisation Engineering Science: Elective Compulsory Logistics and Mobility: Specialisation Traffic Planning and Systems: Elective Compulsory Logistics and Mobility: Specialisation Information Technology: Elective Compulsory Technomathematics: Specialisation I. Mathematics: 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 Information Technology: Elective Compulsory

Course L1046: Graph Theory and Optimization	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Anusch Taraz
Language	DE/EN
Cycle	SoSe
Content	• Graphs, search algorithms for graphs, trees • planar graphs • shortest paths • minimum spanning trees • maximum flow and minimum cut • theorems of Menger, König-Egervary, Hall • NP-complete problems • backtracking and heuristics • linear programming • duality • integer linear programming
Literature	• M. Aigner: Diskrete Mathematik, Vieweg, 2004 • T. Cormen, Ch. Leiserson, R. Rivest, C. Stein: Algorithmen - Eine Einführung, Oldenbourg, 2013 • J. Matousek und J. Nesetril: Diskrete Mathematik, Springer, 2007 • A. Steger: Diskrete Strukturen (Band 1), Springer, 2001 • A. Taraz: Diskrete Mathematik, Birkhäuser, 2012 • V. Turau: Algorithmische Graphentheorie, Oldenbourg, 2009 • K.-H. Zimmermann: Diskrete Mathematik, BoD, 2006

Course L1047: Graph Theory and Optimization	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Anusch Taraz
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	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

Module Responsible	Prof. Karl-Heinz Zimmermann
Admission Requirements	None
Recommended Previous Knowledge	Discrete Algebraic Structures, Automata Theory, Logic, and Formal Language Theory.
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	The students known the important machine models of computability, the class of partial recursive functions, universal computability, Gödel numbering of computations, the theorems of Kleene, Rice, and Rice-Shapiro, the concept of decidable and undecidable sets, the word problems for semi-Thue systems, Thue systems, semi-groups, and Post correspondence systems, Hilbert's 10-th problem, and the basic concepts of complexity theory.
Skills	Students are able to investigate the computability of sets and functions and to analyze the complexity of computable functions.
Personal Competence	
Social Competence	Students are able to solve specific problems alone or in a group and to present the results accordingly.
Autonomy	Students are able to acquire new knowledge from newer literature and to associate the acquired knowledge with other classes.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	60 min
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: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Computer Science: Elective Compulsory Computational Science and Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory

Course L0166: Computability and Complexity Theory

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Karl-Heinz Zimmermann
Language	DE/EN
Cycle	SoSe
Content	
Literature	

Course L0167: Computability and Complexity Theory

Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Karl-Heinz Zimmermann
Language	DE/EN
Cycle	SoSe
Content	
Literature	

Module M0873: Software Industrial Internship

Courses

Title	Typ	Hrs/wk	CP

Module Responsible	Prof. Karl-Heinz Zimmermann
Admission Requirements	None
Recommended Previous Knowledge	Foundations of Software Engineering
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students know the important aspects and phases of software development.
Skills	Students can describe the typical phases of software development and are able to contribute to a software project.
Personal Competence	
Social Competence	Students are able to specify, implement, and analyze specific basic topics in software development and present them accordingly.
Autonomy	Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.
Workload in Hours	Independent Study Time 180, Study Time in Lecture 0
Credit points	6
Course achievement	None
Examination	Written elaboration (accord. to Internship Regulations)
Examination duration and scale	Die Ausarbeitung wird von der Betreuerin bzw. dem Betreuer der Bachelorarbeit bewertet.
Assignment for the Following Curricula	Computer Science: Core Qualification: Compulsory

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	Prof. Karl-Heinz Zimmermann
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	
Knowledge	The students are able to • explicate a specific topic in the field of Computer Science, • describe complex issues, • present different views and evaluate in a critical way.
Skills	The students are able to • familiarize in a specific topic of Computer Science in limited time, • realize a literature survey on the specific topic and cite in a correct way, • elaborate a presentation and give a lecture to a selected audience, • sum up the presentation in 10-15 lines, • answer questions in the final discussion.
Personal Competence	
Social Competence	The students are able to • elaborate and introduce a topic for a certain audience, • discuss the topic, content and structure of the presentation with the instructor, • discuss certain aspects with the audience, and • as the lecturer listen and respond to questions from the audience.
Autonomy	The students are able to • define the task in question in an autonomous way, • develop the necessary knowledge, • use appropriate work equipment, and • guided by an instructor critically check the working status.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Presentation
Examination duration and scale	x
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Core Qualification: Compulsory General Engineering Science (English program, 7 semester): Specialisation Computer Science: Elective Compulsory Computational Science and Engineering: Core Qualification: Compulsory

Course L2362: Introductory Seminar Computer Science I

Typ	Seminar
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Karl-Heinz Zimmermann
Language	DE/EN
Cycle	WiSe/SoSe
Content	
Literature	

Course L2361: Introductory Seminar Computer Science II

Typ	Seminar
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Karl-Heinz Zimmermann
Language	DE/EN
Cycle	WiSe/SoSe
Content	
Literature	

Module M1620: Ethics in Information Technology

Courses

Title	Typ	Hrs/wk	CP
Ethics in Information Technology (L2450)	Lecture	2	3
Ethics in Information Technology (L2451)	Seminar	2	3

Module Responsible	NN
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Subject theoretical and practical work
Examination duration and scale	-
Assignment for the Following Curricula	Computer Science: Core Qualification: Elective Compulsory Data Science: Core Qualification: Compulsory

Course L2450: Ethics in Information Technology

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	NN
Language	DE/EN
Cycle	WiSe
Content	
Literature	Wird zu Beginn der Lehrveranstaltung bekannt gegeben.

Course L2451: Ethics in Information Technology

Typ	Seminar
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	NN
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Specialization I. Computer and Software Engineering

Module M0625: Databases

Courses

Title	Typ	Hrs/wk	CP
Databases (L0337)	Lecture	4	5
Databases (L1150)	Project-/problem-based Learning	1	1

Module Responsible	Prof. Stefan Schulte
Admission Requirements	None
Recommended Previous Knowledge	Students should have basic knowledge in the following areas: • Discrete Algebraic Structures • Procedural Programming • Automata Theory and Formal Languages • Programming Paradigms
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	After successful completion of the course, students know: • Design instruments for relational databases • The relational model • Relational query languages, especially SQL • Requirements on data integrity • Possibilities for query optimization • Aspects of transaction handling, fault handling and concurrency/synchronization in database systems • Specific attributes and differences of object-oriented and object-relational databases • Paradigms and concepts of current technologies for data modelling and database systems
Skills	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.
Personal Competence	
Social Competence	Students can work on complex problems both independently and in teams. They can exchange ideas with each other and use their individual strengths to solve the problem.
Autonomy	Students are able to independently investigate a complex problem and assess which competencies are required to solve it.
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Computer Science: Specialisation Computer and Software Engineering: Elective Compulsory Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Data Science: Core Qualification: Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory

Course L0337: Databases	
Typ	Lecture
Hrs/wk	4
CP	5
Workload in Hours	Independent Study Time 94, Study Time in Lecture 56
Lecturer	Prof. Stefan Schulte
Language	EN
Cycle	WiSe
Content	• Architecture of database systems, conceptual data modeling with the Entity Relationship (ER) modeling language • Relational data model, referential integrity, keys, foreign keys, functional dependencies (FDs), canonical mapping of entity types and relationship into the relational data model, anomalies • Relational algebra as a simple query language • Dependency theory, FD closure, canonical cover of FD set, decomposition of relational schemata, multivalued dependencies, normalization, inclusion dependencies • Practical query languages and integrity constraints w/o considering a conceptual domain model: SQL • Storage structures, database implementation architecture • Index structures • Query processing • Query optimization • Transactions and recovery • Query languages with recursion and consideration of a simple conceptual domain model: Datalog • Semi-naive evaluation strategy, magic sets transformation • Information integration, declarative schema transformation (LAV, GAV), distributed database systems • Description logics, syntax, semantics, decision problems, decision algorithms for Abox satisfiability • Ontology based data access (OBDA), DL-Lite for formalizing ER diagramms • Complexity measure: Data complexity • Semistructured databases and query languages: XML and XQuery
Literature	1. A. Kemper, A. Eickler, Datenbanksysteme - n. Auflage, Oldenbourg, 2010 2. S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, 1995 3. Database Systems, An Application Oriented Approach, Pearson International Edition, 2005 4. H. Garcia-Molina, J.D. Ullman, J. Widom, Database Systems: The Complete Book, Prentice Hall, 2002

Course L1150: Databases	
Typ	Project-/problem-based Learning
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Stefan Schulte
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M0971: Operating Systems

Courses			
Title	**Typ**	**Hrs/wk**	**CP**
Operating Systems (L1153)	Lecture	2	3
Operating Systems (L1154)	Recitation Section (small)	2	3

Module Responsible	Prof. Volker Turau
Admission Requirements	None
Recommended Previous Knowledge	• Object-oriented programming, algorithms, and data structures • Procedural programming • Experience in using tools related to operating systems such as editors, linkers, compilers • Experience in using C-libraries
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students explain the main abstractions process, virtual memory, deadlock, lifelock, and file of operations systems, describe the process states and their transitions, and paraphrase the architectural variants of operating systems. They give examples of existing operating systems and explain their architectures. The participants of the course write concurrent programs using threads, conditional variables and semaphores. Students can describe the variants of realizing a file system. Students explain at least three different scheduling algorithms.
Skills	Students are able to use the POSIX libraries for concurrent programming in a correct and efficient way. They are able to judge the efficiency of a scheduling algorithm for a given scheduling task in a given environment.
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Computer Science: Elective Compulsory Computational Science and Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory

Course L1153: Operating Systems

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Volker Turau
Language	DE
Cycle	SoSe
Content	• Architectures for Operating Systems • Processes • Concurrency • Deadlocks • Memory organization • Scheduling • File systems
Literature	1. Operating Systems, William Stallings, Pearson International Edition 2. Moderne Betriebssysteme, Andrew Tanenbaum, Pearson Studium

Course L1154: Operating Systems

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

Module M1586: Scientific Programming

Courses

Title	**Typ**	**Hrs/wk**	**CP**
Scientific Programming (L2405)	Lecture	3	4
Scientific Programming (L2406)	Recitation Section (small)	2	2

Module Responsible	Prof. Tobias Knopp
Admission Requirements	None
Recommended Previous Knowledge	procedural programming, linear algebra
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	The students • can efficiently solve scientific problems in a modern programming language. • are familiar with the concept of reproducible science. • can handle multidimensional arrays, sparse arrays, data frames and missing data. They know the advantages and disadvantages of specific data structures. • know various ways of presenting data, data relationships and error measures in a suitable way. They are familiar with known data formats for storing scientific data and can select a suitable format for specific data.
Skills	Students are able • to translate complex problems from a mathematical formulation into a suitable program. • to divide a complex problem into subproblems which can be implemented modularly. • to identify numerical standard problems and to use suitable standard algorithms which are available in libraries. • to write maintainable program code, the correctness of which is verified by suitable tests. • to measure the runtime of programs, to identify bottlenecks and to apply suitable acceleration techniques.
Personal Competence	
Social Competence	Students can work on complex problems both independently and in teams. They can exchange ideas with each other and use their individual strengths to solve the problem.
Autonomy	Students are able to independently investigate a complex problem and assess which competencies are required to solve it.
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Data Science: Core Qualification: Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory

Course L2405: Scientific Programming

Typ	Lecture
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Tobias Knopp
Language	DE/EN
Cycle	SoSe
Content	• Elementary Data Types and the Relationship to Mathematics • Scientific data types: Multidimensional Arrays, sparse Arrays, Data Frames, Missing Data • Multiple Dispatch as an Efficient Paradigm for Scientific Programming • Literate Programming • Profiling and benchmarks • Acceleration techniques: caching, multi-threading, SIMD, GPGPU • Scientific data formats: CSV, TOML, HDF5, and selected examples • Data visualization • Standard numerical techniques and efficient program libraries (BLAS, LAPACK, FFTW, ...) • Tests, code management, documentation • Reproducible science
Literature	Ben Lauwens, Allen Downey: Think Julia: How to Think Like a Computer Scientist

Course L2406: Scientific Programming	
Typ	Recitation Section (small)
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Tobias Knopp
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

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

Module Responsible	Prof. Heiko Falk
Admission Requirements	None
Recommended Previous Knowledge	Module "Computer Engineering"
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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.
Skills	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.
Personal Competence	
Social Competence	Students are able to solve similar problems alone or in a group and to present the results accordingly.
Autonomy	Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	**Compulsory** No **Bonus** 15 % **Form** Subject theoretical and practical work **Description**
Examination	Written exam
Examination duration and scale	90 minutes, contents of course and 4 attestations from the PBL "Computer architecture"
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Specialisation Computer and Software Engineering: Elective Compulsory Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Aircraft Systems Engineering: Core Qualification: Elective Compulsory Aircraft Systems Engineering: Specialisation Avionic Systems: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Computer Science: Elective Compulsory Computational Science and Engineering: Specialisation I. Computer Science: Elective Compulsory Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory

Course L0793: Computer Architecture

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Heiko Falk
Language	DE/EN
Cycle	WiSe
Content	• Introduction • VHDL Basics • Programming Models • Realization of Elementary Data Types • Dynamic Scheduling • Branch Prediction • Superscalar Machines • Memory Hierarchies 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.
Literature	• D. Patterson, J. Hennessy. Rechnerorganisation und -entwurf. Elsevier, 2005. • A. Tanenbaum, J. Goodman. Computerarchitektur. Pearson, 2001.

Course L0794: Computer Architecture	
Typ	Project-/problem-based Learning
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Heiko Falk
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Course L1864: Computer Architecture	
Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Heiko Falk
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M0953: Introduction to Information Security

Courses

Title	Typ	Hrs/wk	CP
Introduction to Information Security (L1114)	Lecture	2	3
Introduction to Information Security (L1115)	Recitation Section (small)	2	3

Module Responsible	Prof. Riccardo Scandariato
Admission Requirements	None
Recommended Previous Knowledge	Basics of Computer Science
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students can • name the main security risks when using Information and Communication Systems and name the fundamental security mechanisms, • describe commonly used methods for risk and security analysis, • name the fundamental principles of data protection.
Skills	Students can • evaluate the strenghts and weaknesses of the fundamental security mechanisms and of the commonly used methods for risk and security analysis, • apply the fundamental principles of data protection to concrete cases.
Personal Competence	
Social Competence	Students are capable of appreciating the impact of security problems on those affected and of the potential responsibilities for their resolution.
Autonomy	None
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	120 minutes
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Data Science: Core Qualification: Compulsory

Course L1114: Introduction to Information Security

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Riccardo Scandariato
Language	EN
Cycle	WiSe
Content	• Fundamental concepts • Passwords & biometrics • Introduction to cryptography • Sessions, SSL/TLS • Certificates, electronic signatures • Public key infrastructures • Side-channel analysis • Access control • Privacy • Software security basics • Security management & risk analysis • Security evaluation: Common Criteria
Literature	D. Gollmann: Computer Security, Wiley & Sons, third edition, 2011 Ross Anderson: Security Engineering, Wiley & Sons, second edition, 2008

Course L1115: Introduction to Information Security	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Riccardo Scandariato
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M0803: Embedded Systems

Courses

Title	Typ	Hrs/wk	CP
Embedded Systems (L0805)	Lecture	3	4
Embedded Systems (L0806)	Recitation Section (small)	1	2

Module Responsible	Prof. Heiko Falk
Admission Requirements	None
Recommended Previous Knowledge	Computer Engineering
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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). Another part covers the hardware of embedded systems: Sonsors, 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.
Skills	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.
Personal Competence	
Social Competence	Students are able to solve similar problems alone or in a group and to present the results accordingly.
Autonomy	Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory** **Bonus** **Form** **Description** Yes 10 % Subject theoretical and practical work
Examination	Written exam
Examination duration and scale	90 minutes, contents of course and labs
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory Computer Science: Specialisation Computer and Software Engineering: Elective Compulsory Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Electrical Engineering: Core Qualification: Elective Compulsory Engineering Science: Specialisation Mechatronics: Elective Compulsory Aircraft Systems Engineering: Core Qualification: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechatronics: Elective Compulsory Computational Science and Engineering: Core Qualification: Compulsory Mechatronics: Specialisation System Design: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Mechatronics: Core Qualification: Elective Compulsory Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory

Course L0805: Embedded Systems

Typ	Lecture
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Heiko Falk
Language	EN
Cycle	SoSe
Content	• Introduction • Specifications and Modeling • Embedded/Cyber-Physical Systems Hardware • System Software • Evaluation and Validation • Mapping of Applications to Execution Platforms • Optimization
Literature	• Peter Marwedel. Embedded System Design - Embedded Systems Foundations of Cyber-Physical Systems. 2 nd Edition, Springer, 2012., Springer, 2012.

Course L0806: Embedded Systems	
Typ	Recitation Section (small)
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Heiko Falk
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	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

Module Responsible	Prof. Sibylle Schupp
Admission Requirements	None
Recommended Previous Knowledge	• Practical programming experience • Automata theory and formal languages • Functional programming or procedural programming • Object-oriented programming, algorithms, and data structures • Basic knowledge of software engineering
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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.
Skills	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.
Personal Competence	
Social Competence	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.
Autonomy	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.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Subject theoretical and practical work
Examination duration and scale	Software (Compiler)
Assignment for the Following Curricula	Computer Science: Specialisation Computer and Software Engineering: Elective Compulsory Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computational Science and Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory

Course L0703: Compiler Construction

Typ	Lecture
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	SoSe
Content	• Lexical and syntactic analysis • Semantic analysis • High-level optimization • Intermediate languages and code generation • Compilation pipeline
Literature	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	
Typ	Recitation Section (small)
Hrs/wk	2
CP	4
Workload in Hours	Independent Study Time 92, Study Time in Lecture 28
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	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

Module Responsible	Prof. Sibylle Schupp
Admission Requirements	None
Recommended Previous Knowledge	• Introduction to Software Engineering • Programming Skills • Experience with Developing Small to Medium-Size Programs
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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.
Skills	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
Personal Competence	
Social Competence	Students discuss different design decisions in a group. They defend their solutions orally. They communicate in English.
Autonomy	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, t conduct independent studies to acquire the necessary competencies. They can devise plans to arrive at new solutions or assess existing ones.
Workload in Hours	Independent Study Time 138, Study Time in Lecture 42
Credit points	6
Course achievement	None
Examination	Subject theoretical and practical work
Examination duration and scale	Software
Assignment for the Following Curricula	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science: Specialisation Computer and Software Engineering: Elective Compulsory Computational Science and Engineering: Specialisation I. Computer Science: Elective Compulsory

Course L1790: Software Development	
Typ	Project-/problem-based Learning
Hrs/wk	2
CP	5
Workload in Hours	Independent Study Time 122, Study Time in Lecture 28
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	SoSe
Content	• Agile Methods • Test-Driven Development and Unit Testing • Continuous Integration • Web Services • Scalability • From Defects to Failure
Literature	Duvall, Paul M. Continuous Integration. Pearson Education India, 2007. Humble, Jez, and David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010. Martin, Robert Cecil. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003. http://scrum-kompakt.de/ Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley & Sons, 2011.

Course L1789: Software Development	
Typ	Lecture
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Sibylle Schupp
Language	EN
Cycle	SoSe
Content	• Agile Methods • Test-Driven Development and Unit Testing • Continuous Integration • Web Services • Scalability • From Defects to Failure
Literature	Duvall, Paul M. Continuous Integration. Pearson Education India, 2007. Humble, Jez, and David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010. Martin, Robert Cecil. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003. http://scrum-kompakt.de/ Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley & Sons, 2011.

Specialization II. Mathematics and Engineering Science

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

Module Responsible	Prof. Sabine Le Borne
Admission Requirements	None
Recommended Previous Knowledge	• Mathematik I + II for Engineering Students (german or english) **or** Analysis & Linear Algebra I + II for Technomathematicians • basic MATLAB/Python knowledge
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students are able to • name numerical methods for interpolation, integration, least squares problems, eigenvalue problems, nonlinear root finding problems and to explain their core ideas, • repeat convergence statements for the numerical methods, • explain aspects for the practical execution of numerical methods with respect to computational and storage complexitx.
Skills	Students are able to • implement, apply and compare numerical methods using MATLAB/Python, • justify the convergence behaviour of numerical methods with respect to the problem and solution algorithm, • select and execute a suitable solution approach for a given problem.
Personal Competence	
Social Competence	Students are able to • work together in heterogeneously composed teams (i.e., teams from different study programs and background knowledge), explain theoretical foundations and support each other with practical aspects regarding the implementation of algorithms.
Autonomy	Students are capable • to assess whether the supporting theoretical and practical excercises are better solved individually or in a team, • to assess their individual progess and, if necessary, to ask questions and seek help.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 minutes
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Materials in Engineering Sciences: Compulsory General Engineering Science (German program, 7 semester): Specialisation Biomedical Engineering: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Biomechanics: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Theoretical Mechanical Engineering: Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Aircraft Systems Engineering: Elective Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Mechatronics: Elective Compulsory General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Energy Systems: Elective Compulsory Bioprocess Engineering: Specialisation A - General Bioprocess Engineering: Elective Compulsory Computer Science: Specialisation Computational Mathematics: Elective Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Core Qualification: Compulsory Electrical Engineering: Core Qualification: Elective Compulsory Engineering Science: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory General Engineering Science (English program, 7 semester): Core Qualification: 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 Materials in Engineering Sciences: Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Theoretical Mechanical Engineering: Compulsory

	General Engineering Science (English program, 7 semester): Specialisation Biomedical Engineering: Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechanical Engineering, Focus Mechatronics: Elective Compulsory Computational Science and Engineering: Core Qualification: Compulsory Mechanical Engineering: Specialisation Theoretical Mechanical Engineering: Compulsory Mechanical Engineering: Specialisation Energy Systems: Elective Compulsory Mechanical Engineering: Specialisation Mechatronics: Elective Compulsory Theoretical Mechanical Engineering: Technical Complementary Course Core Studies: Elective Compulsory Process Engineering: Specialisation Process Engineering: Elective Compulsory

Course L0417: Numerical Mathematics I	
Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sabine Le Borne
Language	EN
Cycle	WiSe
Content	1. Finite precision arithmetic, error analysis, conditioning and stability 2. Linear systems of equations: LU and Cholesky factorization, condition 3. Interpolation: polynomial, spline and trigonometric interpolation 4. Nonlinear equations: fixed point iteration, root finding algorithms, Newton's method 5. Linear and nonlinear least squares problems: normal equations, Gram Schmidt and Householder orthogonalization, singular value decomposition, regularizatio, Gauss-Newton and Levenberg-Marquardt methods 6. Eigenvalue problems: power iteration, inverse iteration, QR algorithm 7. Numerical differentiation 8. Numerical integration: Newton-Cotes rules, error estimates, Gauss quadrature, adaptive quadrature
Literature	• Gander/Gander/Kwok: Scientific Computing: An introduction using Maple and MATLAB, Springer (2014) • Stoer/Bulirsch: Numerische Mathematik 1, Springer • Dahmen, Reusken: Numerik für Ingenieure und Naturwissenschaftler, Springer

Course L0418: Numerical Mathematics I	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Sabine Le Borne, Dr. Jens-Peter Zemke
Language	EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1730: Mathematics IV (EN)

Courses

Title	Typ	Hrs/wk	CP
Differential Equations 2 (Partial Differential Equations) (English) (L2783)	Lecture	2	1
Differential Equations 2 (Partial Differential Equations) (English) (L2784)	Recitation Section (large)	1	1
Differential Equations 2 (Partial Differential Equations) (English) (L2785)	Recitation Section (small)	1	1
Complex Functions (English) (L2786)	Lecture	2	1
Complex Functions (English) (L2787)	Recitation Section (large)	1	1
Complex Functions (English) (L2788)	Recitation Section (small)	1	1

Module Responsible	Prof. Anusch Taraz
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Independent Study Time 68, Study Time in Lecture 112
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	120 min
Assignment for the Following Curricula	Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Core Qualification: Elective Compulsory Engineering Science: Core Qualification: Compulsory Engineering Science: Specialisation Electrical Engineering: Compulsory Engineering Science: Specialisation Mechatronics: Elective Compulsory

Course L2783: Differential Equations 2 (Partial Differential Equations) (English)

Typ	Lecture
Hrs/wk	2
CP	1
Workload in Hours	Independent Study Time 2, Study Time in Lecture 28
Lecturer	Dozenten des Fachbereiches Mathematik der UHH
Language	EN
Cycle	SoSe
Content	
Literature	

Course L2784: Differential Equations 2 (Partial Differential Equations) (English)

Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Dozenten des Fachbereiches Mathematik der UHH
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Course L2785: Differential Equations 2 (Partial Differential Equations) (English)

Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Dozenten des Fachbereiches Mathematik der UHH
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Course L2786: Complex Functions (English)	
Typ	Lecture
Hrs/wk	2
CP	1
Workload in Hours	Independent Study Time 2, Study Time in Lecture 28
Lecturer	Dozenten des Fachbereiches Mathematik der UHH
Language	EN
Cycle	SoSe
Content	
Literature	

Course L2787: Complex Functions (English)	
Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Dozenten des Fachbereiches Mathematik der UHH
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Course L2788: Complex Functions (English)	
Typ	Recitation Section (small)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Dozenten des Fachbereiches Mathematik der UHH
Language	EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M0651: Computational Geometry

Courses

Title	**Typ**	**Hrs/wk**	**CP**
Computational Geoemetry (L0393)	Lecture	2	4
Computational Geoemetry (L0394)	Recitation Section (small)	2	2

Module Responsible	Dr. Prashant Batra
Admission Requirements	None
Recommended Previous Knowledge	Linear algebra and analytic geometry as taught in higher secondary school (Computing with vectors a. determinants, Interpretation of scalar product, cross-product, Representation of lines/planes, Satz d. Pythagoras' theorem, cosine theorem, Thales' theorem, projections/embeddings) Basic data structures (trees, binary trees, search trees, balanced binary trees, linked lists) Definition of a graph
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students can name the basic concepts of computer-assisted geometry, describe them with mathematical precision, and explain them by means of examples. Students are conversant with the computational description of geometrical (combinational/topological) facts, including determinant formulas and complexity assessments and proofs for all algorithms, especially output-sensitive algorithms. Students are able to discuss logical connections between these concepts and to explain them by means of examples.
Skills	Students can model tasks from computer-assisted geometry with the aid of the concepts about which they have learnt and can solve them by means of the methods they have learnt.
Personal Competence	
Social Competence	Students are able to discuss with other attendees their own algorithmic suggestions for solving the problems presented. They are also able to work in teams and are conversant with mathematics as a common language.
Autonomy	Students are capable of accessing independently further logical connections between the concepts about which they have learnt and are able to verify them.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Computer Science: Specialisation Computer and Software Engineering: Elective Compulsory Computer Science: Specialisation Computational Mathematics: Elective Compulsory

Course L0393: Computational Geometry	
Typ	Lecture
Hrs/wk	2
CP	4
Workload in Hours	Independent Study Time 92, Study Time in Lecture 28
Lecturer	Dr. Prashant Batra
Language	DE
Cycle	WiSe
Content	Construction of the convex hull of n points, triangulation of a simple polygon Construction of Delaunay-triangulation and Voronoi-diagram Algorithms and data structures for the construction of arrangements, and Ham-Sandwich-Cuts. the intersection of half-planes, the optimization of a linear functional over the latter. Efficiente determination of all intersection of (orthogonal) lines (line segments) Approximative computation of the diameter of a point set Randomised incremental algorithms Basics of lattice point theory , LLL-algorithm and application in integer-valued optimization. Basics of motion planning
Literature	Computational Geometry Algorithms and Applications Authors: • Prof. Dr. Mark de Berg, • Dr. Otfried Cheong, • Dr. Marc van Kreveld, • Prof. Dr. Mark Overmars Springer e-Book: http://dx.doi.org/10.1007/978-3-540-77974-2 Algorithmische Geometrie : Grundlagen, Methoden, Anwendungen / Rolf Klein **Verfasser:** Klein, Rolf **Ausgabe:** 2., vollst. überarb. Aufl. **Erschienen:** Berlin [u.a.] : Springer, 2005 **Umfang:** XI, 392 S. : graph. Darst. Springer e-Book: http://dx.doi.org/10.1007/3-540-27619-X O'Rourke, Joseph Computational geometry in C. (English) Zbl 0816.68124 Cambridge: Univ. Press. ix, 346 p. $ 24.95; £16.95 /sc; $ 59.95; £35.00 /hc (1994). **ISBN:** 0-521-44034-3 ; 0-521-44592-2 **Computational geometry** : an introduction / Franco P. Preparata; Michael Ian Shamos **Verfasser:** Preparata, Franco P. ; Shamos, Michael Ian **Ausgabe:** Corr. and expanded 2. printing. **Erschienen:** New York [u.a.] : Springer, 1988 **Umfang:** XIV, 398 S. : graph. Darst. **Schriftenreihe:** Texts and monographs in computer science **ISBN:** 3-540-96131-3 0-387-96131-3 Devadoss, Satyan L.; O'Rourke, Joseph Discrete and computational geometry. (English) Zbl 1232.52001 Princeton, NJ: Princeton University Press (ISBN 978-0-691-14553-2/hbk; 978-1-400-83898-1/ebook). xi, 255 p. ISBN: 978-3-540-77973-5 (Print) 978-3-540-77974-2 (Online)

Course L0394: Computational Geoemetry	
Typ	Recitation Section (small)
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Dr. Prashant Batra
Language	DE
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

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

Module Responsible	Prof. Anusch Taraz
Admission Requirements	None
Recommended Previous Knowledge	• Mathematics I + II • Discrete Algebraic Structures • Graph Theory and Optimization
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students can name the basic concepts in Combinatorics and Algorithms. They are able to explain them using appropriate examples. • Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples. • They know proof strategies and can reproduce them.
Skills	• Students can model problems in Combinatorics and Algorithms with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods. • Students are able to discover and verify further logical connections between the concepts studied in the course. • For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.
Personal Competence	
Social Competence	• Students are able to work together in teams. They are capable to use mathematics as a common language. • In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.
Autonomy	• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. • Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation Computer and Software Engineering: Elective Compulsory Computer Science: Specialisation Computational Mathematics: Elective Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Core Qualification: Elective Compulsory Computational Science and Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory

Course L1100: Combinatorial Structures and Algorithms	
Typ	Lecture
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Anusch Taraz, Dr. Dennis Clemens
Language	DE/EN
Cycle	WiSe
Content	Counting Structural Graph Theory Analysis of Algorithms Extremal Combinatorics Random discrete structures
Literature	M. Aigner: Diskrete Mathematik, Vieweg, 6. Aufl., 2006 J. Matoušek & J. Nešetřil: Diskrete Mathematik - Eine Entdeckungsreise, Springer, 2007 A. Steger: Diskrete Strukturen - Band 1: Kombinatorik, Graphentheorie, Algebra, Springer, 2. Aufl. 2007 A. Taraz: Diskrete Mathematik, Birkhäuser, 2012.

Course L1101: Combinatorial Structures and Algorithms	
Typ	Recitation Section (small)
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Anusch Taraz
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1242: Quantum Mechanics for Engineers

Courses

Title	Typ	Hrs/wk	CP
Quantum Mechanics for Engineers (L1686)	Lecture	2	3
Quantum Mechanics for Engineers (L1688)	Recitation Section (small)	2	3

Module Responsible	Prof. Wolfgang Hansen
Admission Requirements	None
Recommended Previous Knowledge	• Knowledge in physics, particularly in optics and wave phenomena; • knowledge in mathematics, particularly linear algebra, vector calculus, complex numbers and Fourier expansion
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	The students are able to describe and explain basic terms and principles of quantum mechanics. They can distinguish commons and differences to classical physics and know, in which situations quantum mechanical phenomena may be expected.
Skills	The students get the ability to apply concepts and methods of quantum mechanics to simple problems and systems. Vice versa, they are also able to comprehend requirements and principles of quantum mechanical devices.
Personal Competence	
Social Competence	The students discuss contents of the lectures and present solutions to simple quantum mechanical problems in small groups during the exercises.
Autonomy	The students are able to independently find answers to simple questions on quantum mechanical systems. The students are able to independently comprehend literature to more complex subjects with quantum mechanical background.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	**Compulsory**: No; **Bonus**: None; **Form**: Written elaboration; **Description**: optionale Vorlage von selbst ausgearbeiteten Lösungen zu den Übungen
Examination	Oral exam
Examination duration and scale	90 Minuten
Assignment for the Following Curricula	Computer Science: Specialisation Computational Mathematics: Elective Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Computer Science: Specialisation Computer and Software Engineering: Elective Compulsory Electrical Engineering: Core Qualification: Elective Compulsory

Course L1686: Quantum Mechanics for Engineers

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Wolfgang Hansen
Language	DE
Cycle	WiSe
Content	This lecture introduces into fundamental concepts, methods, and definitions in quantum mechanics, which are needed in modern material and device science. Applications will be discussed using examples in the field of electronic and optical devices. Central topics are: Schrödinger equation, wave function, operators, eigenstates, eigenvalues, quantum wells, harmonic oscillator, tunnel processes, resonant tunnel diode, band structure, density of states, quantum statistics, Zener-diode, stationary perturbation calculation with the quantum-confined Stark effect as an example, Fermi’s golden rule and transition matrix elements, heterostructure laser, quantum cascade laser, many-particle physics, molecules and exchange interaction, quantum bits and quantum cryptography.
Literature	• David J. Griffiths: "Quantenmechanik, eine Einführung", Pearson (2012), ISBN 978-3-8632-6514-4. • David K. Ferry: "Quantum Mechanics", IOP Publishing (1995), ISBN 0-7503-0327-1 (hbk) bzw. 0-7503-0328-X (pbk). • M. Jaros: " Physics and Applications of Semiconductor Microstructures ", Clarendon Press (1989), ISBN: 0-19-851994-X bzw. 0-19-853927-4 (Pbk). • Randy Harris, "Modernes Physik Lehr- und Übungsbuch", 2. aktualisierte Auflage, Kapitel 3-10, Pearson (2013), ISBN 978-3-86894-115-9. • Michael A Nielsen and Isaac L. Chuang: "Quantum Computation and Quantum Informatioin", 10. Auflage, Cambridge University Press (2011), ISBN: 1107002176 9781107002173. • Hiroyuki Sagawa and Nobuaki Yoshida: "Fundamentals of Quantum Information", World Scientific Publishing (2010), ISBN-13: 978-9814324236.

Course L1688: Quantum Mechanics for Engineers	
Typ	Recitation Section (small)
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Wolfgang Hansen
Language	DE
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

Module M1592: Statistics

Courses			
Title	**Typ**	**Hrs/wk**	**CP**
Statistics (L2430)	Lecture	3	4
Statistics (L2431)	Recitation Section (small)	1	2

Module Responsible	Prof. Matthias Schulte
Admission Requirements	None
Recommended Previous Knowledge	Stochastics (oder eine vergleichbare Lehrveranstaltung)
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• Students can name the basic concepts in Statistics. They are able to explain them using appropriate examples. • Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.
Skills	• Students can model statistical problems with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods. They are able to use the statistical software R. • Students are able to discover and verify further logical connections between the concepts studied in the course. • For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.
Personal Competence	
Social Competence	• Students are able to work together (e.g. on their regular home work) in heterogeneously composed teams and to present their results appropriately (e.g. during exercise class). • In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.
Autonomy	• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. • Students can put their knowledge in relation to the contents of other lectures. • Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written exam
Examination duration and scale	90 min
Assignment for the Following Curricula	Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Core Qualification: Compulsory Logistics and Mobility: Specialisation Information Technology: Elective Compulsory Engineering and Management - Major in Logistics and Mobility: Specialisation Information Technology: Elective Compulsory

Course L2430: Statistics

Typ	Lecture
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Matthias Schulte
Language	DE/EN
Cycle	WiSe
Content	• Point estimators • Confidence intervals • Hypothesis testing • Nonparametric statistics • Linear Regression • Time series analysis • Statistical software (R)
Literature	

Course L2431: Statistics	
Typ	Recitation Section (small)
Hrs/wk	1
CP	2
Workload in Hours	Independent Study Time 46, Study Time in Lecture 14
Lecturer	Prof. Matthias Schulte
Language	DE/EN
Cycle	WiSe
Content	See interlocking course
Literature	See interlocking course

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

Module Responsible	Prof. Alexander Schlaefer
Admission Requirements	None
Recommended Previous Knowledge	principles of math (algebra, analysis/calculus) principles of stochastics principles of programming, R/Matlab
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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.
Skills	The students are able to evaluate systems and medical devices in the context of clinical applications.
Personal Competence	
Social Competence	The students describe a problem in medical technology as a project, and define tasks that are solved in a joint effort.
Autonomy	The students can reflect their knowledge and document the results of their work. They can present the results in an appropriate manner.
Workload in Hours	Independent Study Time 110, Study Time in Lecture 70
Credit points	6
Course achievement	**Compulsory** / **Bonus** / **Form** / **Description** Yes / 10 % / Written elaboration Yes / 10 % / Presentation
Examination	Written exam
Examination duration and scale	90 minutes
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Biomedical Engineering: Compulsory Computer Science: Specialisation Computer and Software Engineering: Elective Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: 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 Computational Science and Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Biomedical Engineering: Specialisation Artificial Organs and Regenerative Medicine: Elective Compulsory Biomedical Engineering: Specialisation Implants and Endoprostheses: Elective Compulsory Biomedical Engineering: Specialisation Medical Technology and Control Theory: Elective Compulsory Biomedical Engineering: Specialisation Management and Business Administration: Elective Compulsory Technomathematics: Specialisation III. Engineering Science: Elective Compulsory

Course L0342: Introduction into Medical Technology and Systems

Typ	Lecture
Hrs/wk	2
CP	3
Workload in Hours	Independent Study Time 62, Study Time in Lecture 28
Lecturer	Prof. Alexander Schlaefer
Language	DE
Cycle	SoSe
Content	- imaging systems - computer aided surgery - medical sensor systems - medical information systems - regulatory affairs - standard in medical technology The students will work in groups to apply the methods introduced during the lecture using problem based learning.
Literature	Wird in der Veranstaltung bekannt gegeben.

Course L0343: Introduction into Medical Technology and Systems	
Typ	Project Seminar
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Alexander Schlaefer
Language	DE
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Course L1876: Introduction into Medical Technology and Systems	
Typ	Recitation Section (large)
Hrs/wk	1
CP	1
Workload in Hours	Independent Study Time 16, Study Time in Lecture 14
Lecturer	Prof. Alexander Schlaefer
Language	DE
Cycle	SoSe
Content	- imaging systems - computer aided surgery - medical sensor systems - medical information systems - regulatory affairs - standard in medical technology The students will work in groups to apply the methods introduced during the lecture using problem based learning.
Literature	Wird in der Veranstaltung bekannt gegeben.

Module M0668: Algebra and Control

Courses

Title	Typ	Hrs/wk	CP
Algebra and Control (L0428)	Lecture	2	4
Algebra and Control (L0429)	Recitation Section (small)	2	2

Module Responsible	Dr. Prashant Batra
Admission Requirements	None
Recommended Previous Knowledge	Basics of Real Analysis and Linear Algebra of Vector Spaces and either of: Introduction to Control Theory or: Discrete Mathematics
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students can • Describe input-output systems polynomially • Explain factorization approaches to transfer functions • Name stabilization conditions for systems in coprime stable factorization.
Skills	Students are able to • Undertake a synthesis of stable control loops • Apply suitable methods of analysis and synthesis to describe all stable control loops • Ensure the fulfillment of specified performance measurements.
Personal Competence	
Social Competence	After completing the module, students are able to solve subject-related tasks and to present the results.
Autonomy	Students are provided with tasks which are exam-related so that they can examine their learning progress and reflect on it.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	30 min
Assignment for the Following Curricula	Computer Science: Specialisation Computational Mathematics: Elective Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory

Course L0428: Algebra and Control	
Typ	Lecture
Hrs/wk	2
CP	4
Workload in Hours	Independent Study Time 92, Study Time in Lecture 28
Lecturer	Dr. Prashant Batra
Language	DE/EN
Cycle	SoSe
Content	- Algebraic control methods, polynomial and fractional approach -Single input - single output (SISO) control systems synthesis by algebraic methods, - Simultaneous stabilization - Parametrization of all stabilizing controllers - Selected methods of pole assignment. - Filtering and sensitivity minimization - Polynomial matrices, left and right polynomial fractions. - Euclidean algorithm, diophantine equations over rings - Smith-McMillan normal form - Multiple input - multiple output control system synthesis by polynomial methods, condition of stability.
Literature	• Vidyasagar, M.: Control system synthesis: a factorization approach. The MIT Press,Cambridge/Mass. - London, 1985. • Vardulakis, A.I.G.: Linear multivariable control. Algebraic analysis and synthesis methods, John Wiley & Sons,Chichester,UK,1991. • Chen, Chi-Tsong: Analog and digital control system design. Transfer-function, state-space, and algebraic methods. Oxford Univ. Press,1995. • Kučera, V.: Analysis and Design of Discrete Linear Control Systems. Praha: Academia, 1991.

Course L0429: Algebra and Control	
Typ	Recitation Section (small)
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Dr. Prashant Batra
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Module M1269: Lab Cyber-Physical Systems

Courses

Title	Typ	Hrs/wk	CP
Lab Cyber-Physical Systems (L1740)	Project-/problem-based Learning	4	6

Module Responsible	Prof. Heiko Falk
Admission Requirements	None
Recommended Previous Knowledge	Module "Embedded Systems"
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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. 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, characteristical 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.
Skills	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.
Personal Competence	
Social Competence	Students are able to solve similar problems alone or in a group and to present the results accordingly.
Autonomy	Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Written elaboration
Examination duration and scale	Execution and documentation of all lab experiments
Assignment for the Following Curricula	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Computer Science: Specialisation Computer and Software Engineering: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Computer Science: Elective Compulsory Computational Science and Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Mechatronics: Specialisation Intelligent Systems and Robotics: Elective Compulsory Mechatronics: Specialisation System Design: Elective Compulsory Mechatronics: Technical Complementary Course: Elective Compulsory

Course L1740: Lab Cyber-Physical Systems

Typ	Project-/problem-based Learning
Hrs/wk	4
CP	6
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Lecturer	Prof. Heiko Falk
Language	DE/EN
Cycle	SoSe
Content	• Experiment 1: Programming in NXC • Experiment 2: Programming the Robot in Matlab/Simulink • Experiment 3: Programming the Robot in LabVIEW
Literature	• Peter Marwedel. Embedded System Design - Embedded System Foundations of Cyber-Physical Systems. 2 nd Edition, Springer, 2012. • Begleitende Foliensätze

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	• Mathematics I + II for Engineering students or Analysis & Lineare Algebra I + II for Technomathematicians • Programming experience in C
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	Students can • list classical and modern iteration methods and their interrelationships, • repeat convergence statements for iterative methods, • explain aspects regarding the efficient implementation of iteration methods.
Skills	Students are able to • analyse, implement, test, and compare iterative methods, • analyse the convergence behaviour of iterative methods and, if applicable, compute congergence rates.
Personal Competence	
Social Competence	Students are able to • work together in heterogeneously composed teams (i.e., teams from different study programs and background knowledge), explain theoretical foundations and support each other with practical aspects regarding the implementation of algorithms.
Autonomy	Students are capable • to assess whether the supporting theoretical and practical excercises are better solved individually or in a team, • to work on complex problems over an extended period of time, • to assess their individual progess and, if necessary, to ask questions and seek help.
Workload in Hours	Independent Study Time 124, Study Time in Lecture 56
Credit points	6
Course achievement	None
Examination	Oral exam
Examination duration and scale	20 min
Assignment for the Following Curricula	Computer Science: Specialisation Computational Mathematics: Elective Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Core Qualification: Elective Compulsory Computational Science and Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory

Course L0583: Solvers for Sparse Linear Systems

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

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

Module Responsible	Prof. Gerhard Bauch
Admission Requirements	None
Recommended Previous Knowledge	Mathematics 1-3 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.
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	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.
Skills	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.
Personal Competence	
Social Competence	The students can jointly solve specific problems.
Autonomy	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.
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 Computer Science: Core Qualification: Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Core Qualification: Compulsory Electrical Engineering: Core Qualification: Compulsory Computational Science and Engineering: Core Qualification: Compulsory Mechanical Engineering: Specialisation Mechatronics: Elective Compulsory Mechatronics: Core Qualification: Compulsory Technomathematics: Specialisation III. Engineering Science: Elective Compulsory

Course L0432: Signals and Systems

Typ	Lecture
Hrs/wk	3
CP	4
Workload in Hours	Independent Study Time 78, Study Time in Lecture 42
Lecturer	Prof. Gerhard Bauch
Language	DE/EN
Cycle	SoSe
Content	• Introduction to signal and system theory • Signals ∘ Classification of signals ▪ Continuous-time and discrete-time signals ▪ Analog and digital signals ▪ Deterministic and random signals ∘ Description of LTI systems by differential equations or difference equations, respectively ∘ Basic properties of signals and operations on signals ∘ Elementary signals ∘ Distributions (Generalized Functions) ∘ Power and energy of signals ∘ Correlation functions of deterministic signals ▪ Autocorrelation function ▪ Crosscorrelation function ▪ Orthogonal signals ▪ Applications of correlation • Linear time-invariant (LTI) systems ∘ Linearity ∘ Time-invariance

  - Description of LTI systems by impulse response and frequency response
  - Convolution
  - Convolution and correlation
  - Properties of LTI-systems
  - Causal systems
  - Stable systems
  - Memoryless systems
- Fourier Series and Fourier Transform
    - Fourier transform of continuous-time signals, discrete-time signals, periodic signals, non-periodic signals
    - Properties of the Fourier transform
    - Fourier transform of some basic signals
    - Parseval’s theorem
- Analysis of LTI-systems and signals in the frequency domain
    - Frequency response, magnitude response and phase response
    - Transmission factor, attenuation, gain
    - Frequency-flat and frequency-selective LTI-systems
    - Bandwidth definitions
    - Basic types of systems (filters), lowpass, highpass, bandpass, bandstop systems
    - Phase delay and group delay
    - Linear-phase systems
    - Distortion-free systems
    - Spectrum analysis with limited observation window: Leakage effect
- Laplace Transform
    - Relation of Fourier transform and Laplace transform
    - Properties of the Laplace transform
    - Laplace transform of some basic signals
- Analysis of LTI-systems in the s-domain
    - Transfer function of LTI-systems
    - Relation of Laplace transform, magnitude response and phase response
    - Analysis of LTI-systems using pole-zero plots
    - Allpass filters
    - Minimum-phase, maximum-phase and mixed phase filters
    - Stable systems
- Sampling
    - Sampling theorem
    - Reconstruction of continuous-time signals in frequency domain and time domain
    - Oversampling
    - Aliasing
    - Sampling with pulses of finite duration, sample and hold
    - Decimation and interpolation
- Discrete-Time Fourier Transform (DTFT)
    - Relation of Fourier transform and DTFT
    - Properties of the DTFT
- Discrete Fourier Transform (DFT)
    - Relation of DTFT and DFT
    - Cyclic properties of the DFT
    - DFT matrix
    - Zero padding
    - Cyclic convolution
    - Fast Fourier Transform (FFT)
    - Application of the DFT: Orthogonal Frequency Division Multiplex (OFDM)
- Z-Transform
    - Relation of Laplace transform, DTFT, and z-transform
    - Properties of the z-transform
    - Z-transform of some basic discrete-time signals
- Discrete-time systems, digital filters
    - FIR and IIR filters
    - Z-transform of digital filters
    - Analysis of discrete-time systems using pole-zero plots in the z-domain
    - Stability
    - Allpass filters
    - Minimum-phase, maximum-phase and mixed-phase filters
    - Linear phase filters

Literature

- T. Frey , M. Bossert , Signal- und Systemtheorie, B.G. Teubner Verlag 2004
- K. Kammeyer, K. Kroschel, Digitale Signalverarbeitung, Teubner Verlag.
- B. Girod ,R. Rabensteiner , A. Stenger , Einführung in die Systemtheorie, B.G. Teubner, Stuttgart, 1997
- J.R. Ohm, H.D. Lüke , Signalübertragung, Springer-Verlag 8. Auflage, 2002
- S. Haykin, B. van Veen: Signals and systems. Wiley.
- Oppenheim, A.S. Willsky: Signals and Systems. Pearson.
- Oppenheim, R. W. Schafer: Discrete-time signal processing. Pearson.

Course L0433: Signals and Systems	
Typ	Recitation Section (small)
Hrs/wk	2
CP	2
Workload in Hours	Independent Study Time 32, Study Time in Lecture 28
Lecturer	Prof. Gerhard Bauch
Language	DE/EN
Cycle	SoSe
Content	See interlocking course
Literature	See interlocking course

Specialization III. Subject Specific Focus

Module M1562: Technical Complementary Course I for CSBS

Courses

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

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

Module M1568: Technical Complementary Course II for CSBS

Courses

Title	Typ	Hrs/wk	CP

Module Responsible	Prof. Karl-Heinz Zimmermann
Admission Requirements	None
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	
Skills	
Personal Competence	
Social Competence	
Autonomy	
Workload in Hours	Depends on choice of courses
Credit points	6
Assignment for the Following Curricula	Computer Science: Specialisation 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	• 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.
Recommended Previous Knowledge	
Educational Objectives	After taking part successfully, students have reached the following learning results
Professional Competence	
Knowledge	• 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). • 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. • The students are able to outline the state of research on a selected issue in their subject area.
Skills	• 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. • 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. • The students can take up a critical position on the findings of their own research work from a specialized perspective.
Personal Competence	
Social Competence	• Both in writing and orally the students can outline a scientific issue for an expert audience accurately, understandably and in a structured way. • The students can deal with issues in an expert discussion and answer them in a manner that is appropriate to the addressees. In doing so they can uphold their own assessments and viewpoints convincingly.
Autonomy	• The students are capable of structuring an extensive work process in terms of time and of dealing with an issue within a specified time frame. • The students are able to identify, open up, and connect knowledge and material necessary for working on a scientific problem. • The students can apply the essential techniques of scientific work to research of their own.
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 Computer Science: Thesis: Compulsory Data Science: Thesis: Compulsory Digital Mechanical Engineering: Thesis: Compulsory Electrical Engineering: Thesis: Compulsory Energy and Environmental 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 Computational Science and Engineering: Thesis: Compulsory Logistics and Mobility: Thesis: Compulsory Mechanical Engineering: Thesis: Compulsory Mechatronics: Thesis: Compulsory Naval Architecture: Thesis: Compulsory Technomathematics: Thesis: Compulsory Teilstudiengang Lehramt Elektrotechnik-Informationstechnik: Thesis: Compulsory Teilstudiengang Lehramt Metalltechnik: Thesis: Compulsory Process Engineering: Thesis: Compulsory Engineering and Management - Major in Logistics and Mobility: Thesis: Compulsory