



## **Module Manual**

Bachelor of Science (B.Sc.)

# **Computer Science Dual study program**

Cohort: Winter Term 2022

Updated: 5th August 2024



# 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 M1436: Procedural Programming for Computer Engineers	7
Module M1728: Mathematics I (EN)	9
Module M1755: Linking theory and practice (dual study program, Bachelor's degree)	11
Module M1750: Practical module 1 (dual study program, Bachelor's degree)	13
Module M0624: Automata Theory and Formal Languages	15
Module M0829: Foundations of Management	17
Module M1432: Programming Paradigms	19
Module M1729: Mathematics II (EN)	21
Module M1751: Practical module 2 (dual study program, Bachelor's degree)	23
Module M0834: Computernetworks and Internet Security	25
Module M0625: Databases	27
Module M0730: Computer Engineering	29
Module M1732: Mathematics III (EN)	31
Module M1423: Algorithms and Data Structures	33
Module M1752: Practical module 3 (dual study program, Bachelor's degree)	35
Module M0732: Software Engineering	37
Module M0562: Computability and Complexity Theory	39
Module M0727: Stochastics	41
Module M0852: Graph Theory and Optimization	43
Module M1753: Practical module 4 (dual study program, Bachelor's degree)	45
Module M0873: Software Industrial Internship	47
Module M1578: Seminars Computer Science	48
Module M1754: Practical module 5 (dual study program, Bachelor's degree)	50
Specialization I. Computer and Software Engineering	52
Module M1586: Scientific Programming	52
Module M1595: Machine Learning I	54
Module M1908: Fundamentals of Operating Systems	56
Module M0791: Computer Architecture	58
Module M0953: Introduction to Information Security	60
Module M1593: Data Mining	62
Module M1976: Embedded GPU Projects	64
Module M0754: Compiler Construction	65
Module M1300: Software Development	66
Module M0803: Embedded Systems	68
Specialization II. Mathematics and Engineering Science	70
Module M1730: Mathematics IV (EN)	70
Module M1962: Basics space electronics and primary mission	73
Module M0651: Computational Geometry	74
Module M0941: Combinatorial Structures and Algorithms	77
Module M2046: Introduction to Quantum Computing	79
Module M1592: Statistics	81
Module M0715: Solvers for Sparse Linear Systems	83
Module M0668: Algebra and Control	85
Module M0634: Introduction into Medical Technology and Systems	87
Module M1269: Lab Cyber-Physical Systems	89
Module M0672: Signals and Systems	90
Specialization III. Subject Specific Focus	93
Module M1562: Technical Complementary Course I for CSBS	93
Module M1568: Technical Complementary Course II for CSBS	94
Thesis	95
Module M1800: Bachelor thesis (dual study program)	95

---

---

## **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
<b>Module Responsible</b>	Prof. Karl-Heinz Zimmermann		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Mathematics from High School.		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> The students know the important basics of discrete algebraic structures including elementary combinatorial structures, monoids, groups, rings, fields, finite fields, and vector spaces. They also know specific structures like sub-, sum-, and quotient structures and homomorphisms.</p> <p><i>Skills</i> Students are able to formalize and analyze basic discrete algebraic structures.</p>		
<b>Personal Competence</b>	<p><i>Social Competence</i> Students are able to solve specific problems alone or in a group and to present the results accordingly.</p> <p><i>Autonomy</i> Students are able to acquire new knowledge from specific standard books and to associate the acquired knowledge to other classes.</p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Orientation Studies: Core Qualification: Elective Compulsory		

#### Course L0164: Discrete Algebraic Structures

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

#### Course L0165: Discrete Algebraic Structures

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

Module M0731: Functional Programming				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Functional Programming (L0624)		Lecture	2	2
Functional Programming (L0625)		Recitation Section (large)	2	2
Functional Programming (L0626)		Recitation Section (small)	2	2
<b>Module Responsible</b>	Prof. Sibylle Schupp			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Discrete mathematics at high-school level			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>				
<i>Knowledge</i>	Students apply the principles, constructs, and simple design techniques of functional programming. They demonstrate their ability to read Haskell programs and to explain Haskell syntax as well as Haskell's read-eval-print loop. They interpret warnings and find errors in programs. They apply the fundamental data structures, data types, and type constructors. They employ strategies for unit tests of functions and simple proof techniques for partial and total correctness. They distinguish laziness from other evaluation strategies.			
<i>Skills</i>	Students break a natural-language description down in parts amenable to a formal specification and develop a functional program in a structured way. They assess different language constructs, make conscious selections both at specification and implementations level, and justify their choice. They analyze given programs and rewrite them in a controlled way. They design and implement unit tests and can assess the quality of their tests. They argue for the correctness of their program.			
<b>Personal Competence</b>				
<i>Social Competence</i>	Students practice peer programming with varying peers. They explain problems and solutions to their peer. They defend their programs orally. They communicate in English.			
<i>Autonomy</i>	In programming labs, students learn under supervision (a.k.a. "Betreutes Programmieren") the mechanics of programming. In exercises, they develop solutions individually and independently, and receive feedback.			
<b>Workload in Hours</b>	Independent Study Time 96, Study Time in Lecture 84			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	Yes	15 %	Excercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Elective Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Engineering Science: Specialisation Mechatronics: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechatronics: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory			

Course L0624: Functional Programming	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions</li> <li>• Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions</li> <li>• Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type</li> <li>• Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps)</li> <li>• Modules</li> <li>• Interactive Programming</li> <li>• Lazy Evaluation, Call-by-Value, Strictness</li> <li>• Design Recipes</li> <li>• Testing (axiom-based, invariant-based, against reference implementation)</li> <li>• Reasoning about Programs (equation-based, inductive)</li> <li>• Idioms of Functional Programming</li> <li>• Haskell Syntax and Semantics</li> </ul>
<b>Literature</b>	Graham Hutton, Programming in Haskell, Cambridge University Press 2007.

Course L0625: Functional Programming	
<b>Typ</b>	Recitation Section (large)
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions</li> <li>• Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions</li>   <li>• Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type</li> <li>• Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps)</li> <li>• Modules</li> <li>• Interactive Programming</li> <li>• Lazy Evaluation, Call-by-Value, Strictness</li> <li>• Design Recipes</li> <li>• Testing (axiom-based, invariant-based, against reference implementation)</li> <li>• Reasoning about Programs (equation-based, inductive)</li> <li>• Idioms of Functional Programming</li> <li>• Haskell Syntax and Semantics</li> </ul>
<b>Literature</b>	Graham Hutton, Programming in Haskell, Cambridge University Press 2007.

Course L0626: Functional Programming	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions</li> <li>• Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions</li>   <li>• Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type</li> <li>• Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps)</li> <li>• Modules</li> <li>• Interactive Programming</li> <li>• Lazy Evaluation, Call-by-Value, Strictness</li> <li>• Design Recipes</li> <li>• Testing (axiom-based, invariant-based, against reference implementation)</li> <li>• Reasoning about Programs (equation-based, inductive)</li> <li>• Idioms of Functional Programming</li> <li>• Haskell Syntax and Semantics</li> </ul>
<b>Literature</b>	Graham Hutton, Programming in Haskell, Cambridge University Press 2007.

Module M1436: Procedural Programming for Computer Engineers			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Procedural Programming for Computer Engineers (L2163)		Lecture	2
Procedural Programming for Computer Engineers (L2164)		Recitation Section (large)	1
Procedural Programming for Computer Engineers (L2165)		Practical Course	2
<b>CP</b>			
			2
			1
			3
<b>Module Responsible</b>	Prof. Bernd-Christian Renner		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	None		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	Students will know		
	<ul style="list-style-type: none"> <li>- the essential features of a procedural programming language</li> <li>- the steps during the compilation of procedural source code to machine code</li> <li>- all essential language constructs and data types of a procedural programming language</li> <li>- software design concepts for the implementation of procedural programs</li> </ul>		
<i>Skills</i>	<ul style="list-style-type: none"> <li>- Mastery of typical development tools</li> <li>- Designing simple, structured programs based on a procedural programming language</li> <li>- Debugging by analyzing compiler warnings and error messages</li> <li>- Analysis and explanation of procedural programs</li> </ul>		
<b>Personal Competence</b>			
<i>Social Competence</i>	<ul style="list-style-type: none"> <li>- After completing the module, students are able to work on subject-specific tasks alone or in a group and to present the results appropriately.</li> </ul>		
<i>Autonomy</i>	<ul style="list-style-type: none"> <li>- After completion of the module, students are able to work independently on parts of the subject area using reference books, to summarize the acquired knowledge, to present and to link it with the contents of other courses.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Orientation Studies: Core Qualification: Elective Compulsory Technomathematics: Core Qualification: Compulsory		

Course L2163: Procedural Programming for Computer Engineers	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Bernd-Christian Renner
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>- Development tools: preprocessor, compiler, linker, assembler, IDE, version management (Git)</li> <li>- Procedural programming: fundamental data types, operators, control structures, functions, pointers and arrays, scopes and lifetime of variables, structures / unions, function pointers, Command line arguments</li> <li>- Programming techniques: Modularization, separation of interface and implementation, callback functions, structured data types.</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>- Greg Perry and Dean Miller. C Programming Absolute Beginner's Guide: No experience necessary! Que Publishing; 3. Auflage (7. August 2013). ISBN 978-0789751980.</li> <li>- Helmut Erlenkötter. C: Programmieren von Anfang an. Rowohlt Taschenbuch; 25. Auflage (1. Dezember 1999). ISBN 978-3499600746.</li> <li>- Markus Neumann. C Programmieren: für Einsteiger: Der leichte Weg zum C-Experten (Einfach Programmieren lernen, Band 8). BMU Verlag (30. Januar 2020). ISBN 978-3966450607.</li> <li>- Brian W. Kernighan, Dennis M. Ritchie: The C Programming Language. Prentice Hall; 2. Auflage (1988), ISBN 0-13-110362-8.</li> </ul>



<b>Course L2164: Procedural Programming for Computer Engineers</b>	
<b>Typ</b>	Recitation Section (large)
<b>Hrs/wk</b>	1
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 16, Study Time in Lecture 14
<b>Lecturer</b>	Prof. Bernd-Christian Renner
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

<b>Course L2165: Procedural Programming for Computer Engineers</b>	
<b>Typ</b>	Practical Course
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Bernd-Christian Renner
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1728: Mathematics I (EN)			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Mathematics I (EN) (L2973)	Lecture	4	4
Mathematics I (EN) (L2974)	Recitation Section (large)	2	2
Mathematics I (EN) (L2975)	Recitation Section (small)	2	2
<b>Module Responsible</b>	Prof. Daniel Ruprecht		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	School mathematics		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<ul style="list-style-type: none"> <li>• Students can name the basic concepts in analysis and linear algebra. They are able to explain them using appropriate examples.</li> <li>• Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>• They know proof strategies and can reproduce them.</li> </ul> <ul style="list-style-type: none"> <li>• Students can model problems in analysis and linear algebra with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods.</li> <li>• Students are able to discover and verify further logical connections between the concepts studied in the course.</li> <li>• For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.</li> </ul> <ul style="list-style-type: none"> <li>• Students are able to work together in teams. They are capable to use mathematics as a common language.</li> <li>• In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.</li> </ul> <ul style="list-style-type: none"> <li>• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them.</li> <li>• Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<i>Knowledge</i>			
<i>Skills</i>			
<b>Personal Competence</b>			
<i>Social Competence</i>			
<i>Autonomy</i>			
<b>Workload in Hours</b>	Independent Study Time 128, Study Time in Lecture 112		
<b>Credit points</b>	8		
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b> <b>Description</b>
	Yes	10 %	Excercises
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory		

Course L2973: Mathematics I (EN)	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	4
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 64, Study Time in Lecture 56
<b>Lecturer</b>	Prof. Anusch Taraz
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<p>Mathematical Foundations:</p> <p>sets, statements, induction, mappings, trigonometry</p> <p>Analysis: Foundations of differential calculus in one variable</p> <ul style="list-style-type: none"> <li>• natural and real numbers</li> <li>• convergence of sequences and series</li> <li>• continuous and differentiable functions</li> <li>• mean value theorems</li> <li>• Taylor series</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• T. Arens u.a. : Mathematik, Springer Spektrum, Heidelberg 2015</li> <li>• W. Mackens, H. Voß: Mathematik I für Studierende der Ingenieurwissenschaften, HECO-Verlag, Alsdorf 1994</li> <li>• W. Mackens, H. Voß: Aufgaben und Lösungen zur Mathematik I für Studierende der Ingenieurwissenschaften, HECO-Verlag, Alsdorf 1994</li> <li>• G. Strang: Lineare Algebra, Springer-Verlag, 2003</li> <li>• G. und S. Teschl: Mathematik für Informatiker, Band 1, Springer-Verlag, 2013</li> </ul>

Course L2974: Mathematics I (EN)	
<b>Typ</b>	Recitation Section (large)
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Anusch Taraz, Dr. Dennis Clemens, Dr. Simon Campese
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L2975: Mathematics I (EN)	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Anusch Taraz
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1755: Linking theory and practice (dual study program, Bachelor's degree)	
<b>Module Responsible</b>	Dr. Henning Haschke
<b>Admission Requirements</b>	None
<b>Recommended Previous Knowledge</b>	none
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results
<b>Professional Competence</b> <i>Knowledge</i>	Dual students...  ... can describe and classify selected classic and modern theories, concepts and methods  <ul style="list-style-type: none"> <li>• related to self-management, and organising work and learning</li> <li>• self-competence and</li> <li>• social skills</li> </ul> ... and apply them to specific situations, projects and plans in a personal and professional context.
<i>Skills</i>	Dual students...  <ul style="list-style-type: none"> <li>• ... anticipate typical difficulties, positive and negative effects, as well as success and failure factors in the engineering sector, evaluate them and consider promising strategies and courses of action.</li> </ul>
<b>Personal Competence</b> <i>Social Competence</i>	Dual students...  <ul style="list-style-type: none"> <li>• ... work together in a problem-oriented and interdisciplinary manner as part of expert and work teams.</li> <li>• ... are able to assemble and lead working groups.</li> <li>• ... present complex, subject-related solutions to problems to experts and stakeholders and can develop these further together.</li> </ul>
<i>Autonomy</i>	Dual students...  <ul style="list-style-type: none"> <li>• ... define, reflect and evaluate goals for learning and work processes.</li> <li>• ... design their learning and work processes independently and sustainably at the university and company.</li> <li>• ... take responsibility for their learning and work processes.</li> <li>• ... are able to consciously think through their ideas or actions and relate them to their self-image to develop conclusions for future action based on this.</li> </ul>
<b>Workload in Hours</b>	Independent Study Time 96, Study Time in Lecture 84
<b>Credit points</b>	6
<b>Course achievement</b>	None
<b>Examination</b>	Written elaboration
<b>Examination duration and scale</b>	Studienbegleitende und semesterübergreifende Dokumentation: Die Leistungspunkte für das Modul werden durch die Anfertigung eines digitalen Lern- und Entwicklungsberichtes (E-Portfolio) erworben. Dabei handelt es sich um eine fortlaufende Dokumentation und Reflexion der Lernerfahrungen und der Kompetenzentwicklung im Bereich der Personalen Kompetenz.

Course L2885: Self-Competence for Professional Success in Engineering (for Dual Study Program)	
<b>Typ</b>	Seminar
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Dr. Henning Haschke, Heiko Sieben
<b>Language</b>	DE
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Key qualifications for professional success</li> <li>• Personality and self-image</li> <li>• Personality profiles</li> <li>• Emotional competence</li> <li>• Needs structure models</li> <li>• Motivation theories and models</li> <li>• Communication basics, communication problems</li> <li>• Conflict management</li> <li>• Constructive communication and language cultures</li> <li>• Resilience</li> <li>• Transfer skills and (self-)reflection</li> <li>• Intercultural competence and business etiquette</li> <li>• Documenting and reflecting on learning experiences</li> </ul>
<b>Literature</b>	Seminarapparat

<b>Course L2884: Self-Management, Organising Work and Learning in Engineering (for Dual Study Program)</b>	
<b>Typ</b>	Seminar
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Dr. Henning Haschke, Heiko Sieben
<b>Language</b>	DE
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Learning to learn</li> <li>• Instruments and methods for time and self-management</li> <li>• Personality and work style/behaviour (DISC model); inner drivers/motivation</li> <li>• Goal setting and planning techniques (SMART, GROW); for short-, medium- and long-term planning</li> <li>• Creativity techniques</li> <li>• Stress management, resilience</li> <li>• (Self-)reflection throughout the learning and work process</li> <li>• Structuring/connecting learning and work processes within different learning environments</li> <li>• Factors influencing learning transfer/transfer skills</li> <li>• Documenting and reflecting on learning experiences</li> </ul>
<b>Literature</b>	Seminarapparat

<b>Course L2886: Social-Competence: Team Development and Communication in Engineering (for Dual Study Program)</b>	
<b>Typ</b>	Seminar
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Dr. Henning Haschke, Heiko Sieben
<b>Language</b>	DE
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Forms, conditions and processes of working groups and leadership relationships</li> <li>• Social skills: theories and models</li> <li>• Communication and discussion techniques</li> <li>• Empathy and motivation in teamwork, the way teams work</li> <li>• Critical ability</li> <li>• Team development: ways of developing working and project groups</li> <li>• Insights into day-to-day leadership: theories and models, leadership tasks, leadership styles, situational leadership, basics of change management</li> <li>• Documenting and reflecting on learning experiences</li> </ul>
<b>Literature</b>	Seminarapparat

Module M1750: Practical module 1 (dual study program, Bachelor's degree)			
Courses			
Title	Typ	Hrs/wk	CP
Practical term 1 (dual study program, Bachelor's degree) (L2879)		0	6
<b>Module Responsible</b>	Dr. Henning Haschke		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	A: Self-management, organising work and learning in engineering (for dual study program)		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	Dual students... <ul style="list-style-type: none"> <li>... describe their employer's organisation (company) and the associated regulations that relate to how tasks and competences are distributed, as well as how work processes are handled.</li> <li>... understand the structure and objectives of the dual study programme and the increasing requirements throughout the course of study.</li> </ul>		
<i>Skills</i>	Dual students... <ul style="list-style-type: none"> <li>... use equipment and resources professionally in accordance with the assigned work areas and tasks, and describe operational processes and procedures with regard to the intended work results/objectives.</li> <li>... implement the university's application recommendations in relation to their current tasks.</li> </ul>		
<b>Personal Competence</b>			
<i>Social Competence</i>	Dual students... <ul style="list-style-type: none"> <li>... have familiarised themselves with their new working environment (learning environment) and the associated tasks/processes/working relationships.</li> <li>... know their central points of contact and company colleagues, and exchange ideas with them constructively.</li> <li>... coordinate work tasks with their professional supervisor and ask for support as needed.</li> <li>... help shape the work in the assigned work area and offer their colleagues support to complete their work.</li> <li>... work together with others in smaller work teams in a result-oriented manner.</li> </ul>		
<i>Autonomy</i>	Dual students... <ul style="list-style-type: none"> <li>... structure their work and learning processes within the company independently in line with their responsibilities and authorisations, and coordinate them with their professional supervisor.</li> <li>... complete work tasks/assignments with the support of colleagues.</li> <li>... coordinate the practical phase with any individual preparation required for the examination phase at TUHH.</li> <li>... document and reflect on how their foundational subjects link with their work as an engineer.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 180, Study Time in Lecture 0		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written elaboration		
<b>Examination duration and scale</b>	Documentation accompanying studies and across semesters: Module credit points are earned by completing a digital learning and development report (e-portfolio). This documents and reflects individual learning experiences and skills development relating to interlinking theory and practice, as well as professional practice. In addition, the partner company provides proof to the dual@TUHH Coordination Office that the dual student has completed the practical phase.		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Core Qualification: Compulsory Civil- and Environmental Engineering: Core Qualification: Compulsory Chemical and Bioprocess Engineering: Core Qualification: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Electrical Engineering: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory Green Technologies: Energy, Water, Climate: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Mechanical Engineering: Core Qualification: Compulsory Mechatronics: Core Qualification: Compulsory Naval Architecture: Core Qualification: Compulsory Technomathematics: Core Qualification: Compulsory Engineering and Management - Major in Logistics and Mobility: Core Qualification: Compulsory		

Course L2879: Practical term 1 (dual study program, Bachelor's degree)	
<b>Typ</b>	
<b>Hrs/wk</b>	0
<b>CP</b>	6
<b>Workload in Hours</b>	Independent Study Time 180, Study Time in Lecture 0
<b>Lecturer</b>	Dr. Henning Haschke
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<p><b>Company onboarding process</b></p> <ul style="list-style-type: none"> <li>• Assigning initial work areas (supervisor, colleagues)</li> <li>• Assigning a contact person within the company (usually the HR department)</li> <li>• Assigning a professional mentor in the work area (relating to practical application)</li> <li>• Responsibilities and authorisations of the dual student within the company</li> <li>• Supporting/working with colleagues</li> <li>• Scheduling the relevant practical modules with initial work tasks</li> <li>• Theory/practice transfer options</li> <li>• Scheduling the examination phase/subsequent study semester</li> </ul> <p><b>Operational knowledge and skills</b></p> <ul style="list-style-type: none"> <li>• Company-specific: organisational structure, corporate strategy, business and work areas, work procedures and processes, operational levels</li> <li>• Process and procedure options within the labour-market-relevant field of engineering</li> <li>• Operational equipment and resources</li> <li>• Implementing the university's application recommendations (theory-practice transfer) in corresponding work and task areas across the company</li> </ul> <p><b>Sharing/reflecting on learning</b></p> <ul style="list-style-type: none"> <li>• Creating an e-portfolio</li> <li>• Relevance of foundational subjects when working as an engineer</li> <li>• Comparing the learning and working processes of different learning environments with regard to their results and effects</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Studierendenhandbuch</li> <li>• Betriebliche Dokumente</li> <li>• Hochschulseitige Anwendungsempfehlungen zum Theorie-Praxis-Transfer</li> </ul>

Module M0624: Automata Theory and Formal Languages			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b> <b>CP</b>
Automata Theory and Formal Languages (L0332)		Lecture	2                  4
Automata Theory and Formal Languages (L0507)		Recitation Section (small)	2                  2
<b>Module Responsible</b>	Prof. Matthias Mnich		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	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		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> Students can explain syntax, semantics, and decision problems of propositional logic, and they are able to give algorithms for solving decision problems. Students can show correspondences to Boolean algebra. Students can describe which application problems are hard to represent with propositional logic, and therefore, the students can motivate predicate logic, and define syntax, semantics, and decision problems for this representation formalism. Students can explain unification and resolution for solving the predicate logic SAT decision problem. Students can also describe syntax, semantics, and decision problems for various kinds of temporal logic, and identify their application areas. The participants of the course can define various kinds of finite automata and can identify relationships to logic and formal grammars. The spectrum that students can explain ranges from deterministic and nondeterministic finite automata and pushdown automata to Turing machines. Students can name those formalism for which nondeterminism is more expressive than determinism. They are also able to demonstrate which decision problems require which expressivity, and, in addition, students can transform decision problems w.r.t. one formalism into decision problems w.r.t. other formalisms. They understand that some formalisms easily induce algorithms whereas others are best suited for specifying systems and their properties. Students can describe the relationships between formalisms such as logic, automata, or grammars.</p> <p><i>Skills</i> Students can apply propositional logic as well as predicate logic resolution to a given set of formulas. Students analyze application problems in order to derive propositional logic, predicate logic, or temporal logic formulas to represent them. They can evaluate which formalism is best suited for a particular application problem, and they can demonstrate the application of algorithms for decision problems to specific formulas. Students can also transform nondeterministic automata into deterministic ones, or derive grammars from automata and vice versa. They can show how parsers work, and they can apply algorithms for the language emptiness problem in case of infinite words.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <ul style="list-style-type: none"> <li>• Students are able to work together in teams. They are capable to use mathematics as a common language.</li> <li>• In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.</li> </ul> <p><i>Autonomy</i></p> <ul style="list-style-type: none"> <li>• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them.</li> <li>• Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 min		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Specialisation Mechatronics: Elective Compulsory Engineering Science: Specialisation Mechatronics: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechatronics: Elective Compulsory Computer Science in Engineering: Core Qualification: Compulsory Orientation Studies: Core Qualification: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory		



Course L0332: Automata Theory and Formal Languages	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 92, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Matthias Mnich
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ol style="list-style-type: none"> <li>1. Propositional logic, Boolean algebra, propositional resolution, SAT-2KNF</li> <li>2. Predicate logic, unification, predicate logic resolution</li> <li>3. Temporal Logics (LTL, CTL)</li> <li>4. Deterministic finite automata, definition and construction</li> <li>5. Regular languages, closure properties, word problem, string matching</li> <li>6. Nondeterministic automata: Rabin-Scott transformation of nondeterministic into deterministic automata</li> <li>7. Epsilon automata, minimization of automata, elimination of e-edges, uniqueness of the minimal automaton (modulo renaming of states)</li> <li>8. Myhill-Nerode Theorem: Correctness of the minimization procedure, equivalence classes of strings induced by automata</li> <li>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</li> <li>10. Regular expressions vs. finite automata: Equivalence of formalisms, systematic transformation of representations, reductions</li> <li>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)</li> <li>12. Chomsky normal form</li> <li>13. CYK algorithm for deciding the word problem for context-free grammars</li> <li>14. Deterministic pushdown automata</li> <li>15. Deterministic vs. nondeterministic pushdown automata: Application for parsing, LL(k) or LR(k) grammars and parsers vs. deterministic pushdown automata, compiler compiler</li> <li>16. Regular grammars</li> <li>17. Outlook: Turing machines and linear bounded automata vs general and context-sensitive grammars</li> <li>18. Chomsky hierarchy</li> <li>19. Mealy- and Moore automata: Automata with output (w/o accepting states), infinite state sequences, automata networks</li> <li>20. Omega automata: Automata for infinite input words, Büchi automata, representation of state transition systems, verification w.r.t. temporal logic specifications (in particular LTL)</li> <li>21. LTL safety conditions and model checking with Büchi automata, relationships between automata and logic</li> <li>22. Fixed points, propositional mu-calculus</li> <li>23. Characterization of regular languages by monadic second-order logic (MSO)</li> </ol>
<b>Literature</b>	<ol style="list-style-type: none"> <li>1. Logik für Informatiker Uwe Schöning, Spektrum, 5. Aufl.</li> <li>2. Logik für Informatiker Martin Kreuzer, Stefan Kühling, Pearson Studium, 2006</li> <li>3. Grundkurs Theoretische Informatik, Gottfried Vossen, Kurt-Ulrich Witt, Vieweg-Verlag, 2010.</li> <li>4. Principles of Model Checking, Christel Baier, Joost-Pieter Katoen, The MIT Press, 2007</li> </ol>

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

<b>Module M0829: Foundations of Management</b>				
<b>Courses</b>				
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>	
Management Tutorial (L0882)	Recitation Section (small)	2	3	
Introduction to Management (L0880)	Lecture	3	3	
<b>Module Responsible</b>	Prof. Christoph Ihl			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Basic Knowledge of Mathematics and Business			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	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			
<i>Knowledge</i>	<ul style="list-style-type: none"> <li>explain the differences between Economics and Management and the sub-disciplines in Management and to name important definitions from the field of Management</li> <li>explain the most important aspects of and goals in Management and name the most important aspects of entrepreneurial projects</li> <li>describe and explain basic business functions as production, procurement and sourcing, supply chain management, organization and human resource management, information management, innovation management and marketing</li> <li>explain the relevance of planning and decision making in Business, esp. in situations under multiple objectives and uncertainty, and explain some basic methods from mathematical Finance</li> <li>state basics from accounting and costing and selected controlling methods.</li> </ul>			
<i>Skills</i>	<p>Students are able to analyse business units with respect to different criteria (organization, objectives, strategies etc.) and to carry out an Entrepreneurship project in a team. In particular, they are able to</p> <ul style="list-style-type: none"> <li>analyse Management goals and structure them appropriately</li> <li>analyse organisational and staff structures of companies</li> <li>apply methods for decision making under multiple objectives, under uncertainty and under risk</li> <li>analyse production and procurement systems and Business information systems</li> <li>analyse and apply basic methods of marketing</li> <li>select and apply basic methods from mathematical finance to predefined problems</li> <li>apply basic methods from accounting, costing and controlling to predefined problems</li> </ul>			
<b>Personal Competence</b>	Students are able to			
<i>Social Competence</i>	<ul style="list-style-type: none"> <li>work successfully in a team of students</li> <li>to apply their knowledge from the lecture to an entrepreneurship project and write a coherent report on the project</li> <li>to communicate appropriately and</li> <li>to cooperate respectfully with their fellow students.</li> </ul>			
<i>Autonomy</i>	<p>Students are able to</p> <ul style="list-style-type: none"> <li>work in a team and to organize the team themselves</li> <li>to write a report on their project.</li> </ul>			
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70			
<b>Credit points</b>	6			
<b>Course achievement</b>	None			
<b>Examination</b>	Subject theoretical and practical work			
<b>Examination duration and scale</b>	several written exams during the semester			
<b>Assignment for the Following Curricula</b>	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 Data Science: Core Qualification: Compulsory Electrical Engineering: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Integrated Building Technology: 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	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Christoph Ihl, Katharina Roedelius
<b>Language</b>	DE
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	<p>In the management tutorial, the contents of the lecture will be deepened by practical examples and the application of the discussed tools.</p> <p>If there is adequate demand, a problem-oriented tutorial will be offered in parallel, which students can choose alternatively. Here, students work in groups on selected projects that focus on the elaboration of an innovative business idea from the point of view of an established company or a startup. Again, the business knowledge from the lecture should come to practical use. The group projects are guided by a mentor.</p>
<b>Literature</b>	Relevante Literatur aus der korrespondierenden Vorlesung.

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

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

Course L2169: Programming Paradigms	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Dozenten des SD E
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• fundamentals behind object orientated programming</li> <li>• classes and objects</li> <li>• inheritance (single, multiple)</li> <li>• interfaces</li> <li>• information hiding</li> <li>• exception handling</li> <li>• generic programming and the implementation in the compiler</li> <li>• excursus in programming with dynamically typed programming languages</li> </ul>
<b>Literature</b>	Skript

Course L2170: Programming Paradigms	
<b>Typ</b>	Recitation Section (large)
<b>Hrs/wk</b>	1
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 16, Study Time in Lecture 14
<b>Lecturer</b>	Dozenten des SD E
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• fundamentals behind object orientated programming</li> <li>• classes and objects</li> <li>• inheritance (single, multiple)</li> <li>• interfaces</li> <li>• information hiding</li> <li>• exception handling</li> <li>• generic programming and the implementation in the compiler</li> <li>• excursus in programming with dynamically typed programming languages</li> </ul>
<b>Literature</b>	Skript

Course L2171: Programming Paradigms	
<b>Typ</b>	Practical Course
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Dozenten des SD E
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• fundamentals behind object orientated programming</li> <li>• classes and objects</li> <li>• inheritance (single, multiple)</li> <li>• interfaces</li> <li>• information hiding</li> <li>• exception handling</li> <li>• generic programming and the implementation in the compiler</li> <li>• excursus in programming with dynamically typed programming languages</li> </ul>
<b>Literature</b>	Skript

Module M1729: Mathematics II (EN)				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Mathematics II (EN) (L2979)		Lecture	4	4
Mathematics II (EN) (L2980)		Recitation Section (large)	2	2
Mathematics II (EN) (L2981)		Recitation Section (small)	2	2
<b>Module Responsible</b>	Prof. Daniel Ruprecht			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	School mathematics			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	<ul style="list-style-type: none"> <li>Students can name the basic concepts in analysis and linear algebra. They are able to explain them using appropriate examples.</li> <li>Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>They know proof strategies and can reproduce them.</li> </ul>			
<i>Knowledge</i>				
<i>Skills</i>				
<b>Personal Competence</b>				
<i>Social Competence</i>	<ul style="list-style-type: none"> <li>Students are able to work together in teams. They are capable to use mathematics as a common language.</li> <li>In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.</li> </ul>			
<i>Autonomy</i>	<ul style="list-style-type: none"> <li>Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them.</li> <li>Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>			
<b>Workload in Hours</b>	Independent Study Time 128, Study Time in Lecture 112			
<b>Credit points</b>	8			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	Yes	10 %	Excercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	120 min			
<b>Assignment for the Following Curricula</b>	Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory			

Course L2979: Mathematics II (EN)	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	4
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 64, Study Time in Lecture 56
<b>Lecturer</b>	Prof. Anusch Taraz
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	
<b>Literature</b>	

Course L2980: Mathematics II (EN)	
<b>Typ</b>	Recitation Section (large)
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Anusch Taraz
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

<b>Course L2981: Mathematics II (EN)</b>	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Anusch Taraz
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1751: Practical module 2 (dual study program, Bachelor's degree)			
Courses			
Title	Typ	Hrs/wk	CP
Practical term 2 (dual study program, Bachelor's degree) (L2880)		0	6
<b>Module Responsible</b>	Dr. Henning Haschke		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Successful completion of practical module 1 as part of the dual Bachelor's course</li> <li>• course A from the module on interlinking theory and practice as part of the dual Bachelor's course</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	Dual students ... <ul style="list-style-type: none"> <li>• ... describe their employer's organisational structure (company) and differentiate between associated regulations that relate to how tasks and competences are distributed, as well as how work processes are handled.</li> <li>• ... understand the structure and objectives of the dual study programme and the increasing requirements throughout the course of study.</li> </ul>		
<i>Skills</i>	Dual students ... <ul style="list-style-type: none"> <li>• ... use equipment and resources professionally in accordance with the assigned work areas and tasks, and assess operational processes and procedures with regard to the intended work results/objectives.</li> <li>• ... implement the university's application recommendations in relation to their current tasks.</li> </ul>		
<b>Personal Competence</b>			
<i>Social Competence</i>	Dual students ... <ul style="list-style-type: none"> <li>• ... have familiarised themselves with their new working environment (learning environment) and the associated tasks/processes/working relationships.</li> <li>• ... know their central points of contact and colleagues, and are integrated into the designated tasks and work areas.</li> <li>• ... coordinate work tasks with their professional supervisor and justify procedures and intended results.</li> <li>• ... help shape the work in the assigned work area and offer their colleagues support to complete their work or ask for support based on their needs.</li> <li>• ... work together with others in interdisciplinary work teams in a result-oriented manner.</li> </ul>		
<i>Autonomy</i>	Dual students ... <ul style="list-style-type: none"> <li>• ... structure their work and learning processes within the company independently in line with their responsibilities and authorisations, and coordinate them with their professional supervisor.</li> <li>• ... complete work tasks/assignments independently and/or with the support of colleagues.</li> <li>• ... coordinate the practical phase with any individual preparation required for the examination phase at TUHH.</li> <li>• ... document and reflect on how their foundational subjects link with their work as an engineer.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 180, Study Time in Lecture 0		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written elaboration		
<b>Examination duration and scale</b>	Documentation accompanying studies and across semesters: Module credit points are earned by completing a digital learning and development report (e-portfolio). This documents and reflects individual learning experiences and skills development relating to interlinking theory and practice, as well as professional practice. In addition, the partner company provides proof to the dual@TUHH Coordination Office that the dual student has completed the practical phase.		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Core Qualification: Compulsory Civil- and Environmental Engineering: Core Qualification: Compulsory Chemical and Bioprocess Engineering: Core Qualification: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Electrical Engineering: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory Green Technologies: Energy, Water, Climate: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Mechanical Engineering: Core Qualification: Compulsory Mechatronics: Core Qualification: Compulsory Naval Architecture: Core Qualification: Compulsory Technomathematics: Core Qualification: Compulsory Engineering and Management - Major in Logistics and Mobility: Core Qualification: Compulsory		



Course L2880: Practical term 2 (dual study program, Bachelor's degree)	
<b>Typ</b>	
<b>Hrs/wk</b>	0
<b>CP</b>	6
<b>Workload in Hours</b>	Independent Study Time 180, Study Time in Lecture 0
<b>Lecturer</b>	Dr. Henning Haschke
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<p><b>Company onboarding process</b></p> <ul style="list-style-type: none"> <li>• Assigning work areas (supervisor, colleagues)</li> <li>• Assigning a contact person within the company (usually the HR department)</li> <li>• Assigning a professional mentor in the work area (relating to practical application)</li> <li>• Responsibilities and authorisations of the dual student within the company</li> <li>• Supporting/working with colleagues</li> <li>• Scheduling the relevant practical modules with work tasks</li> <li>• Theory/practice transfer options</li> <li>• Scheduling the examination phase/subsequent study semester</li> </ul> <p><b>Operational knowledge and skills</b></p> <ul style="list-style-type: none"> <li>• Company-specific: organisational structure, corporate strategy, business and work areas, work procedures and processes, operational levels</li> <li>• Process and procedure options within the labour-market-relevant field of engineering</li> <li>• Operational equipment and resources</li> <li>• Implementing the university's application recommendations (theory-practice transfer) in corresponding work and task areas across the company</li> </ul> <p><b>Sharing/reflecting on learning</b></p> <ul style="list-style-type: none"> <li>• Creating an e-portfolio</li> <li>• Relevance of foundational subjects when working as an engineer</li> <li>• Comparing the learning and working processes of different learning environments with regard to their results and effects</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Studierendenhandbuch</li> <li>• Betriebliche Dokumente</li> <li>• Hochschulseitige Anwendungsempfehlungen zum Theorie-Praxis-Transfer</li> </ul>

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
<b>Module Responsible</b>	Prof. Andreas Timm-Giel		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Basics of Computer Science		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> Students are able to explain important and common Internet protocols in detail and classify them, in order to be able to analyse and develop networked systems in further studies and job.</p> <p><i>Skills</i> Students are able to analyse common Internet protocols and evaluate the use of them in different domains.</p>		
<b>Personal Competence</b>	<p><i>Social Competence</i></p> <p><i>Autonomy</i> Students can select relevant parts out of high amount of professional knowledge and can independently learn and understand it.</p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Core Qualification: Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Data Science: Core Qualification: Elective Compulsory Electrical Engineering: Core Qualification: Elective Compulsory Engineering Science: Specialisation Mechatronics: Elective Compulsory Engineering Science: Specialisation Electrical Engineering: Elective Compulsory General Engineering Science (English program, 7 semester): Specialisation Mechatronics: Elective Compulsory Computer Science in Engineering: Core Qualification: Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory		

Course L1098: Computer Networks and Internet Security	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	5
<b>Workload in Hours</b>	Independent Study Time 108, Study Time in Lecture 42
<b>Lecturer</b>	Dr. Koojana Kuladinithi, Prof. Sibylle Fröschele
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<p>In this class an introduction to computer networks with focus on the Internet and its security is given. Basic functionality of complex protocols are introduced. Students learn to understand these and identify common principles. In the exercises these basic principles and an introduction to performance modelling are addressed using computing tasks and physical labs.</p> <p>In the second part of the lecture an introduction to Internet security is given.</p> <p>This class comprises:</p> <ul style="list-style-type: none"> <li>• Introduction to the Internet (TCP/IP model)</li> <li>• Application layer protocols (HTTP, SMTP, DNS)</li> <li>• Transport layer protocols (TCP, UDP)</li> <li>• Network Layer (Internet Protocol IPv4 &amp; IPv6, routing in the Internet)</li> <li>• Data link layer with media access at the example of WLAN</li> <li>• Introduction to Internet Security</li> <li>• Security Aspects of Address Resolution (DNS/DNSSEC, ARP/SEND)</li> <li>• Communication Security (IPSec) - From Address Resolution to Routing (Securing BGP)</li> <li>• Botnets + Firewalls</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Kurose, Ross, Computer Networking - A Top-Down Approach, 8th Edition, Addison-Wesley</li> <li>• Kurose, Ross, Computernetzwerke - Der Top-Down-Ansatz, Pearson Studium; Auflage: 8. Auflage</li> <li>• W. Stallings: Cryptography and Network Security: Principles and Practice, 6th edition</li> </ul> <p>Further literature is announced at the beginning of the lecture.</p>

Course L1099: Computer Networks and Internet Security	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	1
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 16, Study Time in Lecture 14
<b>Lecturer</b>	Dr. Koojana Kuladinithi, Prof. Sibylle Fröschele
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

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

Course L0337: Databases	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 78, Study Time in Lecture 42
<b>Lecturer</b>	Prof. Stefan Schulte
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction to database systems</li> <li>• Design instruments for relational databases, especially entity-relationship</li> <li>• The relational model</li> <li>• Relational query languages, especially SQL</li> <li>• Normalization</li> <li>• Physical data organization</li> <li>• Transaction management</li> <li>• Query optimization</li> <li>• Data representation</li> <li>• Object-oriented and object-relational databases</li> <li>• Paradigms and concepts of current technologies for data modelling and database systems</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• A. Kemper, A. Eickler, Datenbanksysteme, 10. Auflage, De Gruyter, Oldenbourg, 2015</li> <li>• R. Elmasri, S. B. Navathe, Fundamentals of Database Systems, 7th edition, Pearson, 2016</li> </ul>

Course L1150: Databases - Exercise	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Stefan Schulte
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction to database systems</li> <li>• Design instruments for relational databases, especially entity-relationship</li> <li>• The relational model</li> <li>• Relational query languages, especially SQL</li> <li>• Normalization</li> <li>• Physical data organization</li> <li>• Transaction management</li> <li>• Query optimization</li> <li>• Data representation</li> <li>• Object-oriented and object-relational databases</li> <li>• Paradigms and concepts of current technologies for data modelling and database systems</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• A. Kemper, A. Eickler, Datenbanksysteme, 10. Auflage, De Gruyter, Oldenbourg, 2015</li> <li>• R. Elmasri, S. B. Navathe, Fundamentals of Database Systems, 7th edition, Pearson, 2016</li> </ul>

Module M0730: Computer Engineering				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Computer Engineering (L0321)		Lecture	3	4
Computer Engineering (L0324)		Recitation Section (small)	1	2
<b>Module Responsible</b>	Prof. Heiko Falk			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Basic knowledge in electrical engineering			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <p>This module deals with the foundations of the functionality of computing systems. It covers the layers from the assembly-level programming down to gates. The module includes the following topics:</p> <ul style="list-style-type: none"> <li>• Introduction</li> <li>• Combinational logic: Gates, Boolean algebra, Boolean functions, hardware synthesis, combinational networks</li> <li>• Sequential logic: Flip-flops, automata, systematic hardware design</li> <li>• Technological foundations</li> <li>• Computer arithmetic: Integer addition, subtraction, multiplication and division</li> <li>• Basics of computer architecture: Programming models, MIPS single-cycle architecture, pipelining</li> <li>• Memories: Memory hierarchies, SRAM, DRAM, caches</li> <li>• Input/output: I/O from the perspective of the CPU, principles of passing data, point-to-point connections, busses</li> </ul> <p><i>Skills</i></p> <p>The students perceive computer systems from the architect's perspective, i.e., they identify the internal structure and the physical composition of computer systems. The students can analyze, how highly specific and individual computers can be built based on a collection of few and simple components. They are able to distinguish between and to explain the different abstraction layers of today's computing systems - from gates and circuits up to complete processors.</p> <p>After successful completion of the module, the students are able to judge the interdependencies between a physical computer system and the software executed on it. In particular, they shall understand the consequences that the execution of software has on the hardware-centric abstraction layers from the assembly language down to gates. This way, they will be enabled to evaluate the impact that these low abstraction levels have on an entire system's performance and to propose feasible options.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p>Students are able to solve similar problems alone or in a group and to present the results accordingly.</p> <p><i>Autonomy</i></p> <p>Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.</p>			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	Yes	10 %	Excercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 minutes, contents of course and labs			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory General Engineering Science (German program, 7 semester): Specialisation Electrical Engineering: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Elective Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Electrical Engineering: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Integrated Building Technology: Core Qualification: Elective Compulsory Mechatronics: Core Qualification: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory			

Course L0321: Computer Engineering	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 78, Study Time in Lecture 42
<b>Lecturer</b>	Prof. Heiko Falk
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Combinational Logic</li> <li>• Sequential Logic</li> <li>• Technological Foundations</li> <li>• Representations of Numbers, Computer Arithmetics</li> <li>• Foundations of Computer Architecture</li> <li>• Memories</li> <li>• Input/Output</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• A. Clements. The Principles of Computer Hardware. 3. Auflage, Oxford University Press, 2000.</li> <li>• A. Tanenbaum, J. Goodman. Computerarchitektur. Pearson, 2001.</li> <li>• D. Patterson, J. Hennessy. Rechnerorganisation und -entwurf. Elsevier, 2005.</li> </ul>

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

Module M1732: Mathematics III (EN)			
<b>Courses</b>			
Title	Typ	Hrs/wk	CP
Analysis III (EN) (L2790)	Lecture	2	2
Analysis III (EN) (L2791)	Recitation Section (large)	1	1
Analysis III (EN) (L2792)	Recitation Section (small)	1	1
Differential Equations 1 (Ordinary Differential Equations) (EN) (L2793)	Lecture	2	2
Differential Equations 1 (Ordinary Differential Equations) (EN) (L2794)	Recitation Section (large)	1	1
Differential Equations 1 (Ordinary Differential Equations) (EN) (L2795)	Recitation Section (small)	1	1
<b>Module Responsible</b>	Prof. Marko Lindner		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Mathematik I and II (EN or DE)		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<ul style="list-style-type: none"> <li>Students can name the basic concepts in the area of analysis and differential equations. They are able to explain them using appropriate examples.</li> <li>Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>They know proof strategies and can reproduce them.</li> </ul> <ul style="list-style-type: none"> <li>Students can model problems in the area of analysis and differential equations with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods.</li> <li>Students are able to discover and verify further logical connections between the concepts studied in the course.</li> <li>For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.</li> </ul> <ul style="list-style-type: none"> <li>Students are able to work together in teams. They are capable to use mathematics as a common language.</li> <li>In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.</li> </ul> <ul style="list-style-type: none"> <li>Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them.</li> <li>Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<i>Knowledge</i>			
<i>Skills</i>			
<b>Personal Competence</b>			
<i>Social Competence</i>			
<i>Autonomy</i>			
<b>Workload in Hours</b>	Independent Study Time 128, Study Time in Lecture 112		
<b>Credit points</b>	8		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory		

Course L2790: Analysis III (EN)	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	Main features of differential and integrational calculus of several variables <ul style="list-style-type: none"> <li>Differential calculus for several variables</li> <li>Mean value theorems and Taylor's theorem</li> <li>Maximum and minimum values</li> <li>Implicit functions</li> <li>Minimization under equality constraints</li> <li>Newton's method for multiple variables</li> <li>Fourier series</li> <li>Double integrals over general regions</li> <li>Line and surface integrals</li> <li>Theorems of Gauß and Stokes</li> </ul>
<b>Literature</b>	<a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a>



Course L2791: Analysis III (EN)	
<b>Typ</b>	Recitation Section (large)
<b>Hrs/wk</b>	1
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 16, Study Time in Lecture 14
<b>Lecturer</b>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L2792: Analysis III (EN)	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	1
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 16, Study Time in Lecture 14
<b>Lecturer</b>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L2793: Differential Equations 1 (Ordinary Differential Equations) (EN)	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<p>Main features of the theory and numerical treatment of ordinary differential equations</p> <ul style="list-style-type: none"> <li>• Introduction and elementary methods</li> <li>• Existence and uniqueness of initial value problems</li> <li>• Linear differential equations</li> <li>• Stability and qualitative behaviour of the solution</li> <li>• Boundary value problems and basic concepts of calculus of variations</li> <li>• Eigenvalue problems</li> <li>• Numerical methods for the integration of initial and boundary value problems</li> <li>• Classification of partial differential equations</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• <a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a></li> </ul>

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

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

Module M1423: Algorithms and Data Structures				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Algorithms and Data Structures (L2046)		Lecture	4	4
Algorithms and Data Structures (L2047)		Recitation Section (small)	1	2
<b>Module Responsible</b>	Prof. Matthias Mnich			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Discrete Algebraic Structures</li> <li>• Mathematics I</li> <li>• Mathematics II</li> <li>• Procedural Programming</li> <li>• Objectoriented Programming</li> </ul>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	<ul style="list-style-type: none"> <li>• Students can name the basic concepts in algorithm design, algorithm analysis and problem reductions. They are able to explain them using appropriate examples.</li> <li>• Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>• They know proof strategies and can reproduce them.</li> </ul>			
<i>Knowledge</i>				
<i>Skills</i>				
<b>Personal Competence</b>				
<i>Social Competence</i>	<ul style="list-style-type: none"> <li>• Students can model discrete decision, search and optimization problems with the help of the concepts studied in this course. Moreover, they are capable of solving them, and reducing them to each other, by applying established methods.</li> <li>• Students are able to discover and verify further logical connections between the concepts studied in the course.</li> <li>• For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.</li> </ul>			
<i>Autonomy</i>	<ul style="list-style-type: none"> <li>• Students are able to work together in teams. They are capable to use mathematics as a common language.</li> <li>• In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.</li> </ul>			
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	No	20 %	Excercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory General Engineering Science (German program, 7 semester): Specialisation Data Science: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Specialisation Data Science: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Logistics and Mobility: Specialisation Information Technology: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory Engineering and Management - Major in Logistics and Mobility: Specialisation Information Technology: Elective Compulsory			

Course L2046: Algorithms and Data Structures	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	4
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 64, Study Time in Lecture 56
<b>Lecturer</b>	Prof. Matthias Mnich
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Insertion sort</li> <li>• Register machines</li> <li>• Asymptotic analysis, Landau notation</li> <li>• Polynomial-time algorithms and NP-completeness</li> <li>• Divide-and-conquer, merge sort</li> <li>• Strassen algorithm</li> <li>• Greedy algorithm</li> <li>• Dynamic programming</li> <li>• Quick sort</li> <li>• AVL-trees, B-trees</li> <li>• Hashing</li> <li>• Depth first search, breadth first search</li> <li>• Shortest paths</li> <li>• Flow problems, Ford-Fulkerson algorithm</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• T. Cormen, Ch. Leiserson, R. Rivest, C. Stein: Introduction to Algorithms. MIT Press, 2013</li> <li>• S. Skiena: The Algorithm Design Manual. Springer, 2008</li> <li>• J. M. Kleinberg and É. Tardos. Algorithm Design. Addison-Wesley, 2005.</li> </ul>

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

Module M1752: Practical module 3 (dual study program, Bachelor's degree)			
Courses			
Title	Typ	Hrs/wk	CP
Practical term 3 (dual study program, Bachelor's degree) (L2881)			
<b>Module Responsible</b>	Dr. Henning Haschke		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Successful completion of practical module 2 as part of the dual Bachelor's course</li> <li>• course B from the module on interlinking theory and practice as part of the dual Bachelor's course</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	Dual students ... <ul style="list-style-type: none"> <li>• ... understand the company's strategic orientation, as well as the functions and organisation of central departments with their decision-making structures, network relationships.</li> <li>• ... understand the requirements of the engineering profession and correctly estimate the resulting responsibility.</li> <li>• ... combine their knowledge of facts, principles, theories and methods gained from previous study content with acquired practical knowledge - in particular their knowledge of practical professional procedures and approaches, in the current field of activity.</li> </ul>		
<i>Skills</i>	Dual students ... <ul style="list-style-type: none"> <li>• ... apply technical theoretical knowledge to current problems in their own area of work, and evaluate work processes and results.</li> <li>• ... use technology, equipment and resources in accordance with the assigned work areas and tasks, and assess operational processes and procedures with regard to the intended work results/objectives.</li> <li>• ... implement the university's application recommendations in relation to their current tasks.</li> </ul>		
<b>Personal Competence</b>			
<i>Social Competence</i>	Dual students ... <ul style="list-style-type: none"> <li>• ... plan work processes cooperatively, including across work areas.</li> <li>• ... communicate professionally with operational stakeholders and present complex issues in a structured, targeted and convincing manner.</li> </ul>		
<i>Autonomy</i>	Dual students ... <ul style="list-style-type: none"> <li>• ... assume responsibility for work assignments and areas.</li> <li>• ... document and reflect on the relevance of subject modules and specialisations for work as an engineer, as well as the implementation of the university's application recommendations and the associated challenges of a positive transfer of knowledge between theory and practice.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 180, Study Time in Lecture 0		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written elaboration		
<b>Examination duration and scale</b>	Documentation accompanying studies and across semesters: Module credit points are earned by completing a digital learning and development report (e-portfolio). This documents and reflects individual learning experiences and skills development relating to interlinking theory and practice, as well as professional practice. In addition, the partner company provides proof to the dual@TUHH Coordination Office that the dual student has completed the practical phase.		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Core Qualification: Compulsory Civil- and Environmental Engineering: Core Qualification: Compulsory Chemical and Bioprocess Engineering: Core Qualification: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Electrical Engineering: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory Green Technologies: Energy, Water, Climate: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Mechanical Engineering: Core Qualification: Compulsory Mechatronics: Core Qualification: Compulsory Naval Architecture: Core Qualification: Compulsory Technomathematics: Core Qualification: Compulsory Engineering and Management - Major in Logistics and Mobility: Core Qualification: Compulsory		

Course L2881: Practical term 3 (dual study program, Bachelor's degree)	
<b>Typ</b>	
<b>Hrs/wk</b>	0
<b>CP</b>	6
<b>Workload in Hours</b>	Independent Study Time 180, Study Time in Lecture 0
<b>Lecturer</b>	Dr. Henning Haschke
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<p><b>Company onboarding process</b></p> <ul style="list-style-type: none"> <li>• Assigning work area(s)</li> <li>• Extending responsibilities and authorisations of the dual student within the company</li> <li>• Independent work tasks and areas</li> <li>• Participating in project teams</li> <li>• Scheduling the relevant practical modules with work tasks</li> <li>• Theory/practice transfer options</li> <li>• Scheduling the examination phase/subsequent study semester</li> </ul> <p><b>Operational knowledge and skills</b></p> <ul style="list-style-type: none"> <li>• Company-specific: strategic direction, organisation of central business and work areas, departments, decision-making structures, network relationships and internal communication</li> <li>• Linking facts, principles and theories with practical knowledge</li> <li>• Process and procedure options within the labour-market-relevant field of engineering</li> <li>• Operational technology, equipment and resources</li> <li>• Implementing the university's application recommendations (theory-practice transfer) in corresponding work and task areas across the company</li> </ul> <p><b>Sharing/reflecting on learning</b></p> <ul style="list-style-type: none"> <li>• E-portfolio</li> <li>• Relevance of subject modules and specialisations when working as an engineer</li> <li>• University application recommendations for transferring knowledge between theory and practice</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Studierendenhandbuch</li> <li>• Betriebliche Dokumente</li> <li>• Hochschulseitige Anwendungsempfehlungen zum Theorie-Praxis-Transfer</li> </ul>

Module M0732: Software Engineering			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Software Engineering (L0627)	Lecture	2	3
Software Engineering (L0628)	Recitation Section (small)	2	3
<b>Module Responsible</b>	Prof. Sibylle Schupp		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Automata theory and formal languages</li> <li>Procedural programming or Functional programming</li> <li>Object-oriented programming, algorithms, and data structures</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> Students explain the phases of the software life cycle, describe the fundamental terminology and concepts of software engineering, and paraphrase the principles of structured software development. They give examples of software-engineering tasks of existing large-scale systems. They write test cases for different test strategies and devise specifications or models using different notations, and critique both. They explain simple design patterns and the major activities in requirements analysis, maintenance, and project planning.</p> <p><i>Skills</i> For a given task in the software life cycle, students identify the corresponding phase and select an appropriate method. They choose the proper approach for quality assurance. They design tests for realistic systems, assess the quality of the tests, and find errors at different levels. They apply and modify non-executable artifacts. They integrate components based on interface specifications.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> Students practice peer programming. They explain problems and solutions to their peer. They communicate in English.</p> <p><i>Autonomy</i> Using on-line quizzes and accompanying material for self study, students can assess their level of knowledge continuously and adjust it appropriately. Working on exercise problems, they receive additional feedback.</p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>
	Yes	15 %	Exercises
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 min		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Core Qualification: Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory		

Course L0627: Software Engineering	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>Model-based software engineering                             <ul style="list-style-type: none"> <li>Information modeling (use case diagrams)</li> <li>Behavioral modeling (finite state machines, Petri Nets, behavioral UML diagrams)</li> <li>Structural modeling (OOA, UML class diagrams, OCL)</li> <li>Model-based testing</li> </ul> </li> <li>Engineering software products                             <ul style="list-style-type: none"> <li>Agile processes</li> <li>Architecture</li> <li>Code-based testing</li> <li>System-level testing</li> </ul> </li> <li>Software management                             <ul style="list-style-type: none"> <li>Maintenance</li> <li>Project management</li> <li>Software processes</li> </ul> </li> </ul>
<b>Literature</b>	Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, Pearson 2020. Kassem A. Saleh, Software Engineering, J. Ross Publishing 2009.

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

Module M0562: Computability and Complexity Theory				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Computability and Complexity Theory (L0166)		Lecture	2	3
Computability and Complexity Theory (L0167)		Recitation Section (small)	2	3
<b>Module Responsible</b>	Prof. Martin Kliesch			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Discrete Algebraic Structures, Automata Theory, Logic, and Formal Language Theory			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>				
<i>Knowledge</i>	<ul style="list-style-type: none"> <li>• Basic models of computation (finite state machines, Turing machines)</li> <li>• Decision problems and formal languages</li> <li>• Gödel numbering of computations</li> <li>• Universal computability</li> <li>• Decidable and undecidable problems</li> <li>• Reductions, diagonalization, Rice's theorem</li> <li>• Time and space complexity</li> <li>• The complexity classes P and NP</li> <li>• Hierarchy theorems</li> <li>• Polynomial time reductions, NP-completeness</li> <li>• Cook-Levin theorem</li> <li>• Uniform circuit families</li> </ul>			
<i>Skills</i>	After completing this module, students are able to <ul style="list-style-type: none"> <li>• reproduce the knowledge taught in the course,</li> <li>• reproduce simpler proofs of the course and reproduce the ideas of the more complicated ones,</li> <li>• establish connections between the concepts taught, and</li> <li>• apply the learned knowledge to concrete problems.</li> </ul>			
<b>Personal Competence</b>				
<i>Social Competence</i>	After completing this module, students are able to work on subject-specific tasks alone or in a group and to present the results appropriately.			
<i>Autonomy</i>	After completion of this module, students are able to work out sub-areas of the subject area independently on the basis of textbooks and other literature, to summarize and present the acquired knowledge and to link it to the contents of other courses.			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	Yes	15 %	Excercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory General Engineering Science (German program, 7 semester): Specialisation Data Science: Elective Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Elective Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory			

Course L0166: Computability and Complexity Theory	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Martin Kliesch
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	
<b>Literature</b>	



<b>Course L0167: Computability and Complexity Theory</b>	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Martin Kliesch
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0727: Stochastics			
Courses			
Title	Typ	Hrs/wk	CP
Stochastics (L0777)	Lecture	2	4
Stochastics (L0778)	Recitation Section (small)	2	2
<b>Module Responsible</b>	Prof. Matthias Schulte		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Calculus</li> <li>• Discrete algebraic structures (combinatorics)</li> <li>• Propositional logic</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b> <i>Knowledge</i>  <i>Skills</i>  <b>Personal Competence</b> <i>Social Competence</i>  <i>Autonomy</i>	<ul style="list-style-type: none"> <li>• Students can name the basic concepts in Stochastics. They are able to explain them using appropriate examples.</li> <li>• Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>• They know proof strategies and can reproduce them.</li> <li>• Students can model problems from stochastics with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods.</li> <li>• Students are able to discover and verify further logical connections between the concepts studied in the course.</li> <li>• For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.</li> <li>• Students are able to work together (e.g. on their regular home work) in heterogeneously composed teams (i.e., teams from different study programs and background knowledge) and to present their results appropriately (e.g. during exercise class).</li> <li>• In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.</li> <li>• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them.</li> <li>• Students can put their knowledge in relation to the contents of other lectures.</li> <li>• Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory General Engineering Science (German program, 7 semester): Specialisation Advanced Materials: Elective Compulsory General Engineering Science (German program, 7 semester): Specialisation Data Science: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Specialisation Advanced Materials: Elective Compulsory Engineering Science: Specialisation Data Science: Compulsory Engineering Science: Specialisation Electrical Engineering: Elective Compulsory Engineering Science: Specialisation Electrical Engineering: Elective Compulsory Computer Science in Engineering: Core Qualification: Compulsory Logistics and Mobility: Specialisation Information Technology: Elective Compulsory Orientation Studies: Core Qualification: Elective Compulsory Theoretical Mechanical Engineering: Core Qualification: Elective Compulsory Engineering and Management - Major in Logistics and Mobility: Specialisation Information Technology: Elective Compulsory		

Course L0777: Stochastics	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 92, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Matthias Schulte
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Definitions of probability, conditional probability</li> <li>• Random variables</li> <li>• Independence</li> <li>• Distributions and density functions</li> <li>• Characteristics: expectation, variance, standard deviation, moments</li> <li>• Multivariate distributions</li> <li>• Law of large numbers and central limit theorem</li> <li>• Basic notions of stochastic processes</li> <li>• Basic concepts of statistics (point estimators, confidence intervals, hypothesis testing)</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• L. Dümbgen (2003): Stochastik für Informatiker, Springer.</li> <li>• H.-O. Georgii (2012): Stochastics: Introduction to Probability and Statistics, 2nd edition, De Gruyter.</li> <li>• N. Henze (2018): Stochastik für Einsteiger, 12th edition, Springer.</li> <li>• A. Klenke (2014): Probability Theory: A Comprehensive Course, 2nd edition, Springer.</li> <li>• U. Krengel (2005): Einführung in die Wahrscheinlichkeitstheorie und Statistik, 8th edition, Vieweg.</li> <li>• A.N. Shiryaev (2012): Problems in probability, Springer.</li> </ul>

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

Module 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
<b>Module Responsible</b>	Prof. Anusch Taraz		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Discrete Algebraic Structures</li> <li>• Mathematics I</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b> <i>Knowledge</i>	<ul style="list-style-type: none"> <li>• Students can name the basic concepts in Graph Theory and Optimization. They are able to explain them using appropriate examples.</li> <li>• Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>• They know proof strategies and can reproduce them.</li> </ul>		
<i>Skills</i>	<ul style="list-style-type: none"> <li>• 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.</li> <li>• Students are able to discover and verify further logical connections between the concepts studied in the course.</li> <li>• For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.</li> </ul>		
<b>Personal Competence</b> <i>Social Competence</i>	<ul style="list-style-type: none"> <li>• Students are able to work together in teams. They are capable to use mathematics as a common language.</li> <li>• In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.</li> </ul>		
<i>Autonomy</i>	<ul style="list-style-type: none"> <li>• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them.</li> <li>• Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory General Engineering Science (German program, 7 semester): Specialisation Data Science: Elective Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Specialisation Data Science: Elective Compulsory Computer Science in Engineering: Specialisation II. Mathematics & 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	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Anusch Taraz
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Graphs, search algorithms for graphs, trees</li> <li>• planar graphs</li> <li>• shortest paths</li> <li>• minimum spanning trees</li> <li>• maximum flow and minimum cut</li> <li>• theorems of Menger, König-Egervary, Hall</li> <li>• NP-complete problems</li> <li>• backtracking and heuristics</li> <li>• linear programming</li> <li>• duality</li> <li>• integer linear programming</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• M. Aigner: Diskrete Mathematik, Vieweg, 2004</li> <li>• T. Cormen, Ch. Leiserson, R. Rivest, C. Stein: Algorithmen - Eine Einführung, Oldenbourg, 2013</li> <li>• J. Matousek und J. Nešetřil: Diskrete Mathematik, Springer, 2007</li> <li>• A. Steger: Diskrete Strukturen (Band 1), Springer, 2001</li> <li>• A. Taraz: Diskrete Mathematik, Birkhäuser, 2012</li> <li>• V. Turau: Algorithmische Graphentheorie, Oldenbourg, 2009</li> <li>• K.-H. Zimmermann: Diskrete Mathematik, BoD, 2006</li> </ul>

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

Module M1753: Practical module 4 (dual study program, Bachelor's degree)			
Courses			
Title	Typ	Hrs/wk	CP
Practical term 4 (dual study program, Bachelor's degree) (L2882)		0	6
<b>Module Responsible</b>	Dr. Henning Haschke		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Successful completion of practical module 3 as part of the dual Bachelor's course</li> <li>• course B from the module on interlinking theory and practice as part of the dual Bachelor's course</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <p>Dual students ...</p> <ul style="list-style-type: none"> <li>• ... understand the company's strategic orientation, as well as the functions and organisation of central departments with their decision-making structures, network relationships, and relevant company communication.</li> <li>• ... have developed an understanding of the requirements and responsibilities of the engineering profession, know the scope and limits of the professional field of activity.</li> <li>• ... can combine their knowledge of facts, principles, theories and methods gained from previous study content with acquired practical knowledge - in particular their knowledge of practical professional procedures and approaches, in the current field of activity.</li> </ul> <p><i>Skills</i></p> <p>Dual students ...</p> <ul style="list-style-type: none"> <li>• ... apply technical theoretical knowledge to current problems in their own field of work, and evaluate work processes and results, taking into account different possible courses of action.</li> <li>• ... use technology, equipment and resources in accordance with the assigned work areas and tasks, and can assess operational processes and procedures with regard to the intended work results/objectives.</li> <li>• ... implement the university's application recommendations in relation to their current tasks.</li> </ul> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p>Dual students ...</p> <ul style="list-style-type: none"> <li>• ... are able to plan work processes cooperatively, across work areas and in heterogeneous groups.</li> <li>• ... communicate professionally with operational stakeholders and present complex issues in a structured, targeted and convincing manner.</li> </ul> <p><i>Autonomy</i></p> <p>Dual students ...</p> <ul style="list-style-type: none"> <li>• ... assume responsibility for work assignments and areas, and coordinate the associated work processes.</li> <li>• ... document and reflect on the relevance of subject modules and specialisations for work as an engineer, as well as the implementation of the university's application recommendations and the associated challenges of a positive transfer of knowledge between theory and practice.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 180, Study Time in Lecture 0		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written elaboration		
<b>Examination duration and scale</b>	Documentation accompanying studies and across semesters: Module credit points are earned by completing a digital learning and development report (e-portfolio). This documents and reflects individual learning experiences and skills development relating to interlinking theory and practice, as well as professional practice. In addition, the partner company provides proof to the dual@TUHH Coordination Office that the dual student has completed the practical phase.		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Core Qualification: Compulsory Civil- and Environmental Engineering: Core Qualification: Compulsory Chemical and Bioprocess Engineering: Core Qualification: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Electrical Engineering: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory Green Technologies: Energy, Water, Climate: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Mechanical Engineering: Core Qualification: Compulsory Mechatronics: Core Qualification: Compulsory Naval Architecture: Core Qualification: Compulsory Technomathematics: Core Qualification: Compulsory Engineering and Management - Major in Logistics and Mobility: Core Qualification: Compulsory		

Course L2882: Practical term 4 (dual study program, Bachelor's degree)	
<b>Typ</b>	
<b>Hrs/wk</b>	0
<b>CP</b>	6
<b>Workload in Hours</b>	Independent Study Time 180, Study Time in Lecture 0
<b>Lecturer</b>	Dr. Henning Haschke
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<p><b>Company onboarding process</b></p> <ul style="list-style-type: none"> <li>• Assigning work area(s)</li> <li>• Extending responsibilities and authorisations of the dual student within the company</li> <li>• Independent work tasks and areas</li> <li>• Participating in project teams</li> <li>• Scheduling the relevant practical module</li> <li>• Theory/practice transfer options</li> <li>• Scheduling the examination phase/subsequent study semester</li> </ul> <p><b>Operational knowledge and skills</b></p> <ul style="list-style-type: none"> <li>• Company-specific: strategic direction, organisation of central business and work areas, departments, decision-making structures, network relationships and internal communication</li> <li>• Linking facts, principles and theories with practical knowledge</li> <li>• Process and procedure options within the labour-market-relevant field of engineering</li> <li>• Operational technology, equipment and resources</li> <li>• Implementing the university's application recommendations (theory-practice transfer) in corresponding work and task areas across the company</li> </ul> <p><b>Sharing/reflecting on learning</b></p> <ul style="list-style-type: none"> <li>• E-portfolio</li> <li>• Relevance of subject modules and specialisations when working as an engineer</li> <li>• University application recommendations for transferring knowledge between theory and practice</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Studierendenhandbuch</li> <li>• Betriebliche Dokumente</li> <li>• Hochschulseitige Anwendungsempfehlungen zum Theorie-Praxis-Transfer</li> </ul>

Module M0873: Software Industrial Internship			
Courses			
Title	Typ	Hrs/wk	CP
<b>Module Responsible</b>	Dozenten des SD E		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Foundations of Software Engineering		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	Students know the important aspects and phases of software development.		
<i>Skills</i>	Students can describe the typical phases of software development and are able to contribute to a software project.		
<b>Personal Competence</b>			
<i>Social Competence</i>	Students are able to specify, implement, and analyze specific basic topics in software development and present them accordingly.		
<i>Autonomy</i>	Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.		
<b>Workload in Hours</b>	Independent Study Time 180, Study Time in Lecture 0		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written elaboration (accord. to Internship Regulations)		
<b>Examination duration and scale</b>	Die Ausarbeitung wird von der Betreuerin bzw. dem Betreuer der Bachelorarbeit bewertet.		
<b>Assignment for the Following Curricula</b>	Computer Science: Core Qualification: Compulsory		



Module M1578: Seminars Computer Science				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Introductory Seminar Computer Science I (L2362)		Seminar	2	3
Introductory Seminar Computer Science II (L2361)		Seminar	2	3
<b>Module Responsible</b>	Dozenten des SD E			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Basic knowledge of Computer Science and Mathematics at the Bachelor's level.			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>				
<i>Knowledge</i>	The students are able to			
	<ul style="list-style-type: none"> <li>• explicate a specific topic in the field of Computer Science,</li> <li>• describe complex issues,</li> <li>• present different views and evaluate in a critical way.</li> </ul>			
<i>Skills</i>	The students are able to			
	<ul style="list-style-type: none"> <li>• familiarize in a specific topic of Computer Science in limited time,</li> <li>• realize a literature survey on the specific topic and cite in a correct way,</li> <li>• elaborate a presentation and give a lecture to a selected audience,</li> <li>• sum up the presentation in 10-15 lines,</li> <li>• answer questions in the final discussion.</li> </ul>			
<b>Personal Competence</b>				
<i>Social Competence</i>	The students are able to			
	<ul style="list-style-type: none"> <li>• elaborate and introduce a topic for a certain audience,</li> <li>• discuss the topic, content and structure of the presentation with the instructor,</li> <li>• discuss certain aspects with the audience, and</li> <li>• as the lecturer listen and respond to questions from the audience.</li> </ul>			
<i>Autonomy</i>	The students are able to			
	<ul style="list-style-type: none"> <li>• define the task in question in an autonomous way,</li> <li>• develop the necessary knowledge,</li> <li>• use appropriate work equipment, and</li> <li>• guided by an instructor critically check the working status.</li> </ul>			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56			
<b>Credit points</b>	6			
<b>Course achievement</b>	None			
<b>Examination</b>	Presentation			
<b>Examination duration and scale</b>	x			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory General Engineering Science (German program, 7 semester): Specialisation Data Science: Elective Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Specialisation Data Science: Elective Compulsory Engineering Science: Specialisation Information and Communication Systems: Elective Compulsory Computer Science in Engineering: Core Qualification: Compulsory			

Course L2362: Introductory Seminar Computer Science I	
<b>Typ</b>	Seminar
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Dozenten des SD E
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	
<b>Literature</b>	

<b>Course L2361: Introductory Seminar Computer Science II</b>	
<b>Typ</b>	Seminar
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Dozenten des SD E
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	
<b>Literature</b>	

Module M1754: Practical module 5 (dual study program, Bachelor's degree)			
Courses			
Title	Typ	Hrs/wk	CP
Practical term 5 (dual study program, Bachelor's degree) (L2883)			
<b>Module Responsible</b>	Dr. Henning Haschke		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Successful completion of practical module 4 as part of the dual Bachelor's course</li> <li>• course C from the module on interlinking theory and practice as part of the dual Bachelor's course</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	Dual students ... <ul style="list-style-type: none"> <li>• ... combine their knowledge of facts, principles, theories and methods gained from previous study content with acquired practical knowledge - in particular their knowledge of practical professional procedures and approaches, in the current field of activity.</li> <li>• ... have a critical understanding of the practical applications of their engineering subject.</li> </ul>		
<i>Skills</i>	Dual students ... <ul style="list-style-type: none"> <li>• ... apply technical theoretical knowledge to complex, interdisciplinary problems within the company, and evaluate the associated work processes and results, taking into account different possible courses of action.</li> <li>• ... implement the university's application recommendations with regard to their current tasks.</li> <li>• ... develop new solutions as well as procedures and approaches in their field of activity and area of responsibility - including in the case of frequently changing requirements (systemic skills).</li> <li>• ... are able to analyse and evaluate operational issues using academic methods.</li> </ul>		
<b>Personal Competence</b>			
<i>Social Competence</i>	Dual students ... <ul style="list-style-type: none"> <li>• ... work responsibly in operational project teams and proactively deal with problems within their team.</li> <li>• ... represent complex engineering viewpoints, facts, problems and solution approaches in discussions with internal and external stakeholders and develop these further together.</li> </ul>		
<i>Autonomy</i>	Dual students ... <ul style="list-style-type: none"> <li>• ... define goals for their own learning and working processes as engineers.</li> <li>• ... document and reflect on learning and work processes in their area of responsibility.</li> <li>• ... document and reflect on the relevance of subject modules, specialisations and research for work as an engineer, as well as the implementation of the university's application recommendations and the associated challenges of a positive transfer of knowledge between theory and practice.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 180, Study Time in Lecture 0		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written elaboration		
<b>Examination duration and scale</b>	Documentation accompanying studies and across semesters: Module credit points are earned by completing a digital learning and development report (e-portfolio). This documents and reflects individual learning experiences and skills development relating to interlinking theory and practice, as well as professional practice. In addition, the partner company provides proof to the dual@TUHH Coordination Office that the dual student has completed the practical phase.		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Core Qualification: Compulsory Civil- and Environmental Engineering: Core Qualification: Compulsory Chemical and Bioprocess Engineering: Core Qualification: Compulsory Computer Science: Core Qualification: Compulsory Data Science: Core Qualification: Compulsory Electrical Engineering: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory Green Technologies: Energy, Water, Climate: Core Qualification: Compulsory Computer Science in Engineering: Core Qualification: Compulsory Mechanical Engineering: Core Qualification: Compulsory Mechatronics: Core Qualification: Compulsory Naval Architecture: Core Qualification: Compulsory Technomathematics: Core Qualification: Compulsory Engineering and Management - Major in Logistics and Mobility: Core Qualification: Compulsory		

Course L2883: Practical term 5 (dual study program, Bachelor's degree)	
<b>Typ</b>	
<b>Hrs/wk</b>	0
<b>CP</b>	6
<b>Workload in Hours</b>	Independent Study Time 180, Study Time in Lecture 0
<b>Lecturer</b>	Dr. Henning Haschke
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<p><b>Company onboarding process</b></p> <ul style="list-style-type: none"> <li>• Assigning a future professional field of activity as an engineer (B.Sc.) and associated areas of work</li> <li>• Extending responsibilities and authorisations of the dual student within the company up to the intended first assignment after completing their studies or to the assignment completed during the subsequent dual Master's course</li> <li>• Taking personal responsibility within a team - in their own area of responsibility and across departments</li> <li>• Scheduling the final practical module with a clear correlation to work structures</li> <li>• Internal agreement on a potential topic for the Bachelor's dissertation</li> <li>• Planning the Bachelor's dissertation within the company in cooperation with TU Hamburg</li> <li>• Scheduling the examination phase/sixth study semester</li> </ul> <p><b>Operational knowledge and skills</b></p> <ul style="list-style-type: none"> <li>• Company-specific: dealing with change, team development, responsibility as an engineer in their own future field of work (B.Sc.), dealing with complex contexts and unresolved problems, developing and implementing innovative solutions</li> <li>• Specialising in one field of work (final dissertation)</li> <li>• Systemic skills</li> <li>• Implementing the university's application recommendations (theory-practice transfer) in corresponding work and task areas across the company</li> </ul> <p><b>Sharing/reflecting on learning</b></p> <ul style="list-style-type: none"> <li>• E-portfolio</li> <li>• Relevance of subject modules and specialisations when working as an engineer</li> <li>• Importance of research and innovation when working as an engineer</li> <li>• University application recommendations for transferring knowledge between theory and practice</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Studierendenhandbuch</li> <li>• Betriebliche Dokumente</li> <li>• Hochschuleitige Anwendungsempfehlungen zum Theorie-Praxis-Transfer</li> </ul>

## Specialization I. Computer and Software Engineering

Module M1586: Scientific Programming			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Scientific Programming (L2405)	Lecture	3	4
Scientific Programming (L2406)	Recitation Section (small)	2	2
<b>Module Responsible</b>	Prof. Tobias Knopp		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	procedural programming, linear algebra		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <p>The students</p> <ul style="list-style-type: none"> <li>• can efficiently solve scientific problems in a modern programming language.</li> <li>• are familiar with the concept of reproducible science.</li> <li>• can handle multidimensional arrays, sparse arrays, data frames and missing data. They know the advantages and disadvantages of specific data structures.</li> <li>• 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.</li> </ul> <p><i>Skills</i></p> <p>Students are able</p> <ul style="list-style-type: none"> <li>• to translate complex problems from a mathematical formulation into a suitable program.</li> <li>• to divide a complex problem into subproblems which can be implemented modularly.</li> <li>• to identify numerical standard problems and to use suitable standard algorithms which are available in libraries.</li> <li>• to write maintainable program code, the correctness of which is verified by suitable tests.</li> <li>• to measure the runtime of programs, to identify bottlenecks and to apply suitable acceleration techniques.</li> </ul> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p>Students can work on complex problems both independently and in teams. They can exchange ideas with each other and use their individual strengths to solve the problem.</p> <p><i>Autonomy</i></p> <p>Students are able to independently investigate a complex problem and assess which competencies are required to solve it.</p>		
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Subject theoretical and practical work		
<b>Examination duration and scale</b>	exercise task, group project with presentation, and written test		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Data Science: Elective Compulsory Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Specialisation Data Science: Elective Compulsory Mechatronics: Specialisation Dynamic Systems and AI: Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory		

Course L2405: Scientific Programming	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 78, Study Time in Lecture 42
<b>Lecturer</b>	Prof. Tobias Knopp
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Elementary Data Types and the Relationship to Mathematics</li> <li>• Scientific data types: Multidimensional Arrays, sparse Arrays, Data Frames, Missing Data</li> <li>• Multiple Dispatch as an Efficient Paradigm for Scientific Programming</li> <li>• Literate Programming</li> <li>• Profiling and benchmarks</li> <li>• Acceleration techniques: caching, multi-threading, SIMD, GPGPU</li> <li>• Scientific data formats: CSV, TOML, HDF5, and selected examples</li> <li>• Data visualization</li> <li>• Standard numerical techniques and efficient program libraries (BLAS, LAPACK, FFTW, ...)</li> <li>• Tests, code management, documentation</li> <li>• Reproducible science</li> </ul>
<b>Literature</b>	Ben Lauwens, Allen Downey: Think Julia: How to Think Like a Computer Scientist

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

Module M1595: Machine Learning I			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b> <b>CP</b>
Machine Learning I (L2432)		Lecture	2                  3
Machine Learning I (L2433)		Recitation Section (small)	3                  3
<b>Module Responsible</b>	Prof. Nihat Ay		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Linear Algebra, Analysis, Basic Programming Course		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <p>The students know</p> <ul style="list-style-type: none"> <li>• general principles of machine learning learning: supervised/unsupervised learning, generative/descriptive learning, parametric/non-parametric learning</li> <li>• different learning methods: neural networks, support vector machines, clustering, dimensionality reduction, kernel methods</li> <li>• fundamentals of statistical learning theory</li> <li>• advanced techniques such as transfer learning, reinforcement learning, generative adversarial networks and adaptive control</li> </ul> <p><i>Skills</i></p> <p>The students can</p> <ul style="list-style-type: none"> <li>• apply machine learning methods to concrete problems</li> <li>• select and evaluate suitable methods for specific problems</li> <li>• evaluate the quality of a trained data-driven model</li> <li>• work with known software frameworks for machine learning</li> <li>• adapt the architecture and cost function of neural networks to specific problems</li> <li>• show the limits of machine learning methods</li> </ul> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p>Students can work on complex problems both independently and in teams. They can exchange ideas with each other and use their individual strengths to solve the problem.</p> <p><i>Autonomy</i></p> <p>Students are able to independently investigate a complex problem and assess which competencies are required to solve it.</p>		
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70		
<b>Credit points</b>	6		
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b> <b>Description</b>
	No	20 %	Excercises
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 min		
<b>Assignment for the Following Curricula</b>	<p>General Engineering Science (German program, 7 semester): Specialisation Mechanical Engineering, Focus Theoretical Mechanical Engineering: Elective Compulsory</p> <p>General Engineering Science (German program, 7 semester): Specialisation Data Science: Compulsory</p> <p>Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory</p> <p>Data Science: Core Qualification: Compulsory</p> <p>Engineering Science: Specialisation Advanced Materials: Elective Compulsory</p> <p>Engineering Science: Specialisation Mechatronics: Elective Compulsory</p> <p>Engineering Science: Specialisation Data Science: Compulsory</p> <p>Engineering Science: Specialisation Mechanical Engineering: Elective Compulsory</p> <p>Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory</p> <p>Logistics and Mobility: Specialisation Information Technology: Elective Compulsory</p> <p>Mechanical Engineering: Specialisation Theoretical Mechanical Engineering: Elective Compulsory</p> <p>Mechatronics: Specialisation Dynamic Systems and AI: Compulsory</p> <p>Technomathematics: Specialisation II. Informatics: Elective Compulsory</p> <p>Engineering and Management - Major in Logistics and Mobility: Specialisation Information Technology: Elective Compulsory</p>		

Course L2432: Machine Learning I	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Nihat Ay
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• History of neuroscience and machine learning (in particular, the age of deep learning)</li> <li>• McCulloch-Pitts neurons and binary Artificial Neural Networks</li> <li>• Boolean and threshold functions</li> <li>• Universality of McCulloch-Pitts neural networks</li> <li>• Learning and the perceptron convergence theorem</li> <li>• Support vector machines</li> <li>• Harmonic analysis of Boolean functions</li> <li>• Continuous Artificial Neural Networks</li> <li>• Kolmogorov's superposition theorem</li> <li>• Universal approximation with continuous neural networks</li> <li>• Approximation error and the gradient decent method: the general idea</li> <li>• The stochastic gradient decent method (Robbins-Monro and Kiefer-Wolfowitz cases)</li> <li>• Multilayer networks and the backpropagation algorithm</li> <li>• Statistical Learning Theory</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Martin Anthony and Peter L. Bartlett. Neural Network Learning: Theoretical Foundations. Cambridge University Press, 1999.</li> <li>• Martin Anthony. Discrete Mathematics of Neural Networks: Selected Topics. SIAM Monographs on Discrete Mathematics &amp; Applications, 1987.</li> <li>• Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar. Foundations of Machine Learning, Second Edition. MIT Press, 2018.</li> <li>• Christopher M. Bishop. Pattern Recognition and Machine Learning. Information Science and Statistics. Springer-Verlag, 2008.</li> <li>• Bernhard Schölkopf, Alexander Smola. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Adaptive Computation and Machine Learning series. MIT Press, Cambridge, MA, 2002.</li> <li>• Luc Devroye, László Györfi, Gábor Lugosi. A Probabilistic Theory of Pattern Recognition. Springer, 1996.</li> <li>• Vladimir Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag: New York, Berlin, Heidelberg, 1995.</li> </ul>

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



Module M1908: Fundamentals of Operating Systems			
Courses			
Title	Typ	Hrs/wk	CP
Fundamentals of Operating Systems (L3148)	Lecture	2	3
Fundamentals of Operating Systems (L3149)	Recitation Section (small)	2	3
<b>Module Responsible</b>	Prof. Christian Dietrich		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Procedural programming in C, as well as associated tools (editor, linker, compiler)</li> <li>• Foundations of computer architecture</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> The course provides basic knowledge about the structure, functionality and system-level use of operating systems. Using the model of a multi-level machine, students learn about operating system abstractions such as processes, threads, virtual memory, files, device files and inter-process communication, as well as techniques for their efficient implementation. This includes strategies for process scheduling, latency minimization through buffering, and main and background memory management. Furthermore, they know the topics of security in the operating system context and aspects of system-oriented software development in C. In the lecture-accompanying exercises, they deepened material practically on the basis programming tasks in C from the range of the UNIX system programming. The students are familiar with the operating system functions for single-processor systems. They have become familiar with special issues relating to multiprocessor systems (based on shared memory) in passing and in relation to functions for coordinating concurrent programs. Similarly, they know the topic of real-time processing to some extent only in relation to process scheduling.</p> <p><i>Skills</i> Students will be able to use the POSIX system interface to access the various resources of the computing system. They are able to grasp technical documentation in order to implement complex interaction protocols. They are able to recognize concurrency problems and avoid them with blocking synchronization primitives.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> Students are able to discuss and collaboratively present a problem in small groups with reference to operating systems and systems software.</p> <p><i>Autonomy</i> Students are able to independently prepare and review the lecture content.</p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 min		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory		

Course L3148: Fundamentals of Operating Systems	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Christian Dietrich
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Basic OS concepts</li> <li>• System-oriented software development in C</li> <li>• Files and file systems</li> <li>• Processes and threads</li> <li>• Interrupts, system calls and signals</li> <li>• Process scheduling</li> <li>• Memory based interaction</li> <li>• Resource management, synchronization and jamming</li> <li>• Inter-process communication</li> <li>• Memory organization</li> <li>• Storage virtualization</li> <li>• System security and access protection</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Operating Systems. Internals and Design Principles; William Stallings; Prentice Hall 2008; ISBN: 978-0136006329.</li> <li>• Operating System Concepts; Abraham Silberschatz, Greg Gagne, Peter Bear Galvin; John Wiley &amp; Sons, Inc.; 2005 ISBN: 0-471-69466-5.</li> <li>• Modern Operating Systems; Andrew S. Tanenbaum; Prentice Hall 2007 ISBN: 978-0136006633</li> <li>• Structured Computer Organization; Andrew S. Tanenbaum; Prentice Hall 2006 ISBN: 978-0131485211.</li> </ul>

Course L3149: Fundamentals of Operating Systems	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Christian Dietrich
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0791: Computer Architecture				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Computer Architecture (L0793)		Lecture	2	3
Computer Architecture (L0794)		Project-/problem-based Learning	2	2
Computer Architecture (L1864)		Recitation Section (small)	1	1
<b>Module Responsible</b>	Prof. Heiko Falk			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Module "Computer Engineering"			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>				
<i>Knowledge</i>	This module presents advanced concepts from the discipline of computer architecture. In the beginning, a broad overview over various programming models is given, both for general-purpose computers and for special-purpose machines (e.g., signal processors). Next, foundational aspects of the micro-architecture of processors are covered. Here, the focus particularly lies on the so-called pipelining and the methods used for the acceleration of instruction execution used in this context. The students get to know concepts for dynamic scheduling, branch prediction, superscalar execution of machine instructions and for memory hierarchies.			
<i>Skills</i>	The students are able to describe the organization of processors. They know the different architectural principles and programming models. The students examine various structures of pipelined processor architectures and are able to explain their concepts and to analyze them w.r.t. criteria like, e.g., performance or energy efficiency. They evaluate different structures of memory hierarchies, know parallel computer architectures and are able to distinguish between instruction- and data-level parallelism.			
<b>Personal Competence</b>				
<i>Social Competence</i>	Students are able to solve similar problems alone or in a group and to present the results accordingly.			
<i>Autonomy</i>	Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.			
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	No	15 %	Subject	theoretical and practical work
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 minutes, contents of course and 4 attestations from the PBL "Computer architecture"			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Aircraft Systems Engineering: Core Qualification: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Aeronautics: Core Qualification: Elective Compulsory Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory			

Course L0793: Computer Architecture	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Heiko Falk
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• VHDL Basics</li> <li>• Programming Models</li> <li>• Realization of Elementary Data Types</li> <li>• Dynamic Scheduling</li> <li>• Branch Prediction</li> <li>• Superscalar Machines</li> <li>• Memory Hierarchies</li> </ul> <p>The theoretical tutorials amplify the lecture's content by solving and discussing exercise sheets and thus serve as exam preparation. Practical aspects of computer architecture are taught in the FPGA-based PBL on computer architecture whose attendance is mandatory.</p>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• D. Patterson, J. Hennessy. Rechnerorganisation und -entwurf. Elsevier, 2005.</li> <li>• A. Tanenbaum, J. Goodman. Computerarchitektur. Pearson, 2001.</li> </ul>

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

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

Module M0953: Introduction to Information Security				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Introduction to Information Security (L1114)		Lecture	2	3
Introduction to Information Security (L1115)		Recitation Section (small)	2	3
<b>Module Responsible</b>	Prof. Riccardo Scandariato			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Basics of Computer Science			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>				
<i>Knowledge</i>	Students can <ul style="list-style-type: none"> <li>name the main security risks when using Information and Communication Systems,</li> <li>name the fundamental security mechanisms,</li> <li>name the fundamental principles of data protection.</li> </ul>			
<i>Skills</i>	Students can <ul style="list-style-type: none"> <li>evaluate the strenghts and weaknesses of the fundamental security mechanisms,</li> <li>apply the fundamental principles of data protection to concrete cases.</li> </ul>			
<b>Personal Competence</b>				
<i>Social Competence</i>	Students are capable of appreciating the impact of security problems on those affected and of the potential responsibilities for their resolution.			
<i>Autonomy</i>	None			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	No	5 %	Subject theoretical and practical work	andGruppenarbeit mit aktuellen Technologien aus dem Bereich Sicherheit
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	120 minutes			
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Specialisation Information and Communication Systems: Compulsory			

Course L1114: Introduction to Information Security	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Riccardo Scandariato
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>Fundamental concepts</li> <li>Passwords &amp; biometrics, Single-Sign-On</li> <li>Passwordless authentication</li> <li>Introduction to cryptography</li> <li>Certificates, electronic signatures</li> <li>Public key infrastructures</li> <li>Sessions, TLS</li> <li>Access control</li> <li>Privacy</li> <li>Software security basics</li> </ul>
<b>Literature</b>	Ross Anderson: Security Engineering, Wiley & Sons, 3rd edition, 2020

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

Module M1593: Data Mining				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Data Mining (L2434)		Lecture	2	3
Data Mining (L2435)		Project-/problem-based Learning	2	3
<b>Module Responsible</b>	Prof. Stefan Schulte			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Databases</li> <li>Machine learning</li> </ul>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	After successful completion of the course, students know:			
<i>Knowledge</i>	<ul style="list-style-type: none"> <li>Basic concepts for data preparation</li> <li>Similarity and distance measures</li> <li>Methods to mine data patterns</li> <li>Procedures to analyse clusters</li> <li>Approaches to identify outliers</li> <li>Data mining for different types of data, e.g., data streams, text data, time series data</li> </ul>			
<i>Skills</i>	Students are able to analyze large, heterogeneous volumes of data. They know methods and their application to recognize patterns in data sets and data clusters. The students are able to apply the studied methods in different domains, e.g., for data streams, text data, or time series data.			
<b>Personal Competence</b>	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.			
<i>Social Competence</i>				
<i>Autonomy</i>	Students are able to independently investigate a complex problem and assess which competencies are required to solve it.			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	Yes	20 %	Subject theoretical and practical work	andPraktische Arbeiten zu bestimmten Themen aus dem Bereich Data Mining
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Data Science: Compulsory Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Specialisation Data Science: Compulsory Logistics and Mobility: Specialisation Information Technology: Elective Compulsory Mechatronics: Specialisation Dynamic Systems and AI: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory Engineering and Management - Major in Logistics and Mobility: Specialisation II. Information Technology: Elective Compulsory			

Course L2434: Data Mining	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Stefan Schulte, Dr. Dominik Schallmoser
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>Data preparation</li> <li>Similarity and distance measures</li> <li>Pattern mining</li> <li>Cluster analysis</li> <li>Outliers detection</li> <li>Data mining for different types of data, e.g., data streams, text data, time series data</li> </ul>
<b>Literature</b>	Charu C. Aggarwal: Text Mining - The Textbook, Springer, 2015. Available at <a href="https://link.springer.com/book/10.1007/978-3-319-14142-8">https://link.springer.com/book/10.1007/978-3-319-14142-8</a>

Course L2435: Data Mining	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Stefan Schulte, Dr. Dominik Schallmoser
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course



Module M1976: Embedded GPU Projects			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Embedded GPU Projects (L3224)		Project-/problem-based Learning	4
			<b>CP</b>
			6
<b>Module Responsible</b>	Prof. Sohan Lal		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	An introductory module on computer engineering or computer architecture, and good programming skills in Python/C++.		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	Students will learn the architecture and organization of heterogenous embedded platforms consisting of multi-core CPUs and many-core GPUs such as Jetson boards from NVIDIA. Students will program them using Python/CUDA/C++. Depending upon the specific requirements of projects, students will learn the deployment of various deep learning frameworks, such as TensorFlow, and PyTorch, on embedded platforms. In addition, students will also develop expertise in various domains such as space applications and autonomous driving.		
<i>Skills</i>	By the end of this module, students will have mastered the intricacies of embedded boards encompassing GPUs especially Jetson boards from NVIDIA and gained invaluable experience in real-world project development (using various deep learning frameworks) within the dynamic fields of space and autonomous driving.		
<b>Personal Competence</b>			
<i>Social Competence</i>	By participating in team projects, students gain a holistic set of soft and social skills, including communication skills, and collaboration in teamwork, that not only contribute to their academic success but also prepare them for success in their future careers and personal interactions.		
<i>Autonomy</i>	Students learn to take ownership of individual and collective tasks within the team; fulfilling commitments and delivering quality work on time.		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written elaboration		
<b>Examination duration and scale</b>	Report on achieved results		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory		

Course L3224: Embedded GPU Projects	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	4
<b>CP</b>	6
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56
<b>Lecturer</b>	Prof. Sohan Lal
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<p>This is a project-based learning module where students will work with embedded boards encompassing GPUs, such as Jetson boards from NVIDIA to work on cutting-edge projects from various domains such as space and autonomous driving. An introduction to embedded boards including a set of projects will be proposed at the beginning of the module. It is also possible for a student to propose his/her project in consultation with the lecturer. The students will be encouraged to work in small teams (1-3 team members). The team-based approach not only facilitates a supportive learning environment but also equips students with the ability to tackle complex problems synergistically.</p> <p>By the end of this module, students will have mastered the intricacies of embedded boards encompassing GPUs and gained invaluable experience in real-world project development within the dynamic fields of space and autonomous driving.</p>
<b>Literature</b>	<ol style="list-style-type: none"> <li>Kosmidis et al., "GPU4S: Embedded GPUs in Space," Euromicro Conference on Digital System Design (DSD), 2019, pp. 399-405, doi: 10.1109/DSD.2019.00064.</li> <li>Prashanthi et al., "Characterizing the Performance of Accelerated Jetson Edge Devices for Training Deep Learning Models," Proc. ACM Meas. Anal. Comput. Syst., 2022</li> <li>Lim et al., "Onboard Artificial Intelligence for Space Situational Awareness with Low-Power GPUs," AMOS, 2020</li> <li>Hsueh et al., "Fault Injection Techniques and Tools," In: Computer, Vol. 30, No. 4, pp. 75-82, 1997</li> <li>Hari et al., "SASSIFI: An Architecture-Level Fault Injection Tool for GPU Application Resilience Evaluation," In: International Symposium on Performance Analysis of Systems and Software (ISPASS), USA, 2017</li> <li>Jha et al., "ML-Based Fault Injection for Autonomous Vehicles: A Case for Bayesian Fault Injection," In: International Conference on Dependable Systems and Networks (DSN), USA, 2019</li> <li>Ziaja et al., "Benchmarking Deep Learning for On-Board Space Applications," In: Remote Sensing, 2021</li> <li>"Towards a European AI4EO R&amp;I Agenda" <a href="https://phiweek2018.esa.int/agenda/files/session58.pdf">https://phiweek2018.esa.int/agenda/files/session58.pdf</a></li> </ol>

Module M0754: Compiler Construction			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Compiler Construction (L0703)		Lecture	2
Compiler Construction (L0704)		Recitation Section (small)	2
<b>CP</b>			4
<b>Module Responsible</b>	Prof. Sibylle Schupp		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Practical programming experience</li> <li>• Automata theory and formal languages</li> <li>• Functional programming or procedural programming</li> <li>• Object-oriented programming, algorithms, and data structures</li> <li>• Basic knowledge of software engineering</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> Students explain the workings of a compiler and break down a compilation task in different phases. They apply and modify the major algorithms for compiler construction and code improvement. They can re-write those algorithms in a programming language, run and test them. They choose appropriate internal languages and representations and justify their choice. They explain and modify implementations of existing compiler frameworks and experiment with frameworks and tools.</p> <p><i>Skills</i> Students design and implement arbitrary compilation phases. They integrate their code in existing compiler frameworks. They organize their compiler code properly as a software project. They generalize algorithms for compiler construction to algorithms that analyze or synthesize software.</p>		
<b>Personal Competence</b>	<p><i>Social Competence</i> Students develop the software in a team. They explain problems and solutions to their team members. They present and defend their software in class. They communicate in English.</p> <p><i>Autonomy</i> Students develop their software independently and define milestones by themselves. They receive feedback throughout the entire project. They organize the software project so that they can assess their progress themselves.</p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Subject theoretical and practical work		
<b>Examination duration and scale</b>	Software (Compiler)		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory		

Course L0703: Compiler Construction	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	2
<b>Workload in Hours</b>	Independent Study Time 32, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Lexical and syntactic analysis</li> <li>• Semantic analysis</li> <li>• High-level optimization</li> <li>• Intermediate languages and code generation</li> <li>• Compilation pipeline</li> </ul>
<b>Literature</b>	Alfred Aho, Jeffrey Ullman, Ravi Sethi, and Monica S. Lam, Compilers: Principles, Techniques, and Tools, 2nd edition  Aarne Ranta, Implementing Programming Languages, An Introduction to Compilers and Interpreters, with an appendix coauthored by Markus Forsberg, College Publications, London, 2012

Course L0704: Compiler Construction	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 92, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1300: Software Development			
Courses			
Title	Typ	Hrs/wk	CP
Software Development (L1790)	Project-/problem-based Learning	2	5
Software Development (L1789)	Lecture	1	1
<b>Module Responsible</b>	Prof. Sibylle Schupp		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Introduction to Software Engineering</li> <li>• Programming Skills</li> <li>• Experience with Developing Small to Medium-Size Programs</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <p>Students explain the fundamental concepts of agile methods, describe the process of test-driven development, and explain how continuous integration can be used in different scenarios. They give examples of selected pitfalls in software development, regarding scalability and other non-functional requirements. They write unit tests and build scripts and combine them in a corresponding integration environment. They explain major activities in requirements analysis, program comprehension, and agile project development.</p> <p><i>Skills</i></p> <p>For a given task on a legacy system, students identify the corresponding parts in the system and select an appropriate method for understanding the details. They choose the proper approach of splitting a task in independent testable and extensible pieces and, thus, solve the task with proper methods for quality assurance. They design tests for legacy systems, create automated builds, and find errors at different levels. They integrate the resulting artifacts in a continuous development environment</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p>Students discuss different design decisions in a group. They defend their solutions orally. They communicate in English.</p> <p><i>Autonomy</i></p> <p>Using accompanying tools, students can assess their level of knowledge continuously and adjust it appropriately. Within limits, they can set their own goals. Upon successful completion, students can identify and formulate concrete problems of software systems and propose solutions. Within this field, they can conduct independent studies to acquire the necessary competencies. They can devise plans to arrive at new solutions or assess existing ones.</p>		
<b>Workload in Hours</b>	Independent Study Time 138, Study Time in Lecture 42		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Subject theoretical and practical work		
<b>Examination duration and scale</b>	Software		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory		

Course L1790: Software Development	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	2
<b>CP</b>	5
<b>Workload in Hours</b>	Independent Study Time 122, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Agile Methods</li> <li>• Test-Driven Development and Unit Testing</li> <li>• Continuous Integration</li> <li>• Web Services</li> <li>• Scalability</li> <li>• From Defects to Failure</li> </ul>
<b>Literature</b>	<p>Duvall, Paul M. Continuous Integration. Pearson Education India, 2007.</p> <p>Humble, Jez, and David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010.</p> <p>Martin, Robert Cecil. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003.</p> <p><a href="http://scrum-kompakt.de/">http://scrum-kompakt.de/</a></p> <p>Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley &amp; Sons, 2011.</p>

Course L1789: Software Development	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	1
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 16, Study Time in Lecture 14
<b>Lecturer</b>	Prof. Sibylle Schupp
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Agile Methods</li> <li>• Test-Driven Development and Unit Testing</li> <li>• Continuous Integration</li> <li>• Web Services</li> <li>• Scalability</li> <li>• From Defects to Failure</li> </ul>
<b>Literature</b>	<p>Duvall, Paul M. Continuous Integration. Pearson Education India, 2007.</p> <p>Humble, Jez, and David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010.</p> <p>Martin, Robert Cecil. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003.</p> <p><a href="http://scrum-kompakt.de/">http://scrum-kompakt.de/</a></p> <p>Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley &amp; Sons, 2011.</p>

Module M0803: Embedded Systems				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Embedded Systems (L0805)		Lecture	3	3
Embedded Systems (L2938)		Project-/problem-based Learning	1	1
Embedded Systems (L0806)		Recitation Section (small)	1	2
<b>Module Responsible</b>	Prof. Heiko Falk			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Computer Engineering			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	<p><i>Knowledge</i> Embedded systems can be defined as information processing systems embedded into enclosing products. This course teaches the foundations of such systems. In particular, it deals with an introduction into these systems (notions, common characteristics) and their specification languages (models of computation, hierarchical automata, specification of distributed systems, task graphs, specification of real-time applications, translations between different models).</p> <p>Another part covers the hardware of embedded systems: Sensors, A/D and D/A converters, real-time capable communication hardware, embedded processors, memories, energy dissipation, reconfigurable logic and actuators. The course also features an introduction into real-time operating systems, middleware and real-time scheduling. Finally, the implementation of embedded systems using hardware/software co-design (hardware/software partitioning, high-level transformations of specifications, energy-efficient realizations, compilers for embedded processors) is covered.</p> <p><i>Skills</i> After having attended the course, students shall be able to realize simple embedded systems. The students shall realize which relevant parts of technological competences to use in order to obtain a functional embedded systems. In particular, they shall be able to compare different models of computations and feasible techniques for system-level design. They shall be able to judge in which areas of embedded system design specific risks exist.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> Students are able to solve similar problems alone or in a group and to present the results accordingly.</p> <p><i>Autonomy</i> Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.</p>			
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	Yes	10 %	Subject	theoretical and practical work
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 minutes, contents of course and labs			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Compulsory Computer Science: Specialisation I. Computer and Software Engineering: Elective Compulsory Electrical Engineering: Core Qualification: Elective Compulsory Engineering Science: Specialisation Electrical Engineering: Elective Compulsory Engineering Science: Specialisation Information and Communication Systems: 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 Computer Science in Engineering: Core Qualification: Compulsory Aeronautics: Core Qualification: Elective Compulsory Mechatronics: Core Qualification: Elective Compulsory Mechatronics: Specialisation Naval Engineering: Compulsory Mechatronics: Specialisation Electrical Systems: Compulsory Mechatronics: Specialisation Dynamic Systems and AI: Compulsory Mechatronics: Specialisation Robot- and Machine-Systems: Compulsory Mechatronics: Specialisation Medical Engineering: Compulsory Microelectronics and Microsystems: Specialisation Embedded Systems: Elective Compulsory			

Course L0805: Embedded Systems	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 48, Study Time in Lecture 42
<b>Lecturer</b>	Prof. Heiko Falk
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Specifications and Modeling</li> <li>• Embedded/Cyber-Physical Systems Hardware</li> <li>• System Software</li> <li>• Evaluation and Validation</li> <li>• Mapping of Applications to Execution Platforms</li> <li>• Optimization</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Peter Marwedel. Embedded System Design - Embedded Systems Foundations of Cyber-Physical Systems. 2<sup>nd</sup> Edition, Springer, 2012., Springer, 2012.</li> </ul>

Course L2938: Embedded Systems	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	1
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 16, Study Time in Lecture 14
<b>Lecturer</b>	Prof. Heiko Falk
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Specifications and Modeling</li> <li>• Embedded/Cyber-Physical Systems Hardware</li> <li>• System Software</li> <li>• Evaluation and Validation</li> <li>• Mapping of Applications to Execution Platforms</li> <li>• Optimization</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Peter Marwedel. Embedded System Design - Embedded Systems Foundations of Cyber-Physical Systems. 2<sup>nd</sup> Edition, Springer, 2012., Springer, 2012.</li> </ul>

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

## Specialization II. Mathematics and Engineering Science

### Module M1730: Mathematics IV (EN)

Courses			
Title	Typ	Hrs/wk	CP
Differential Equations 2 (Partial Differential Equations) (EN) (L2783)	Lecture	2	1
Differential Equations 2 (Partial Differential Equations) (EN) (L2784)	Recitation Section (large)	1	1
Differential Equations 2 (Partial Differential Equations) (EN) (L2785)	Recitation Section (small)	1	1
Complex Functions (EN) (L2786)	Lecture	2	1
Complex Functions (EN) (L2787)	Recitation Section (large)	1	1
Complex Functions (EN) (L2788)	Recitation Section (small)	1	1
<b>Module Responsible</b>	Prof. Marko Lindner		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Mathematics I - III (EN or DE)		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<ul style="list-style-type: none"> <li>• Students can name the basic concepts in Mathematics IV. They are able to explain them using appropriate examples.</li> <li>• Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>• They know proof strategies and can reproduce them.</li> </ul> <ul style="list-style-type: none"> <li>• Students can model problems in Mathematics IV with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods.</li> <li>• Students are able to discover and verify further logical connections between the concepts studied in the course.</li> <li>• For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.</li> </ul> <ul style="list-style-type: none"> <li>• Students are able to work together in teams. They are capable to use mathematics as a common language.</li> <li>• In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.</li> </ul> <ul style="list-style-type: none"> <li>• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them.</li> <li>• Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<i>Knowledge</i>			
<i>Skills</i>			
<b>Personal Competence</b>			
<i>Social Competence</i>			
<i>Autonomy</i>			
<b>Workload in Hours</b>	Independent Study Time 68, Study Time in Lecture 112		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	120 min		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Advanced Materials: Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Core Qualification: Elective Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Engineering Science: Core Qualification: Compulsory Engineering Science: Core Qualification: Compulsory Engineering Science: Specialisation Advanced Materials: Compulsory Engineering Science: Specialisation Mechatronics: Compulsory Engineering Science: Specialisation Biomedical Engineering: Compulsory Engineering Science: Specialisation Electrical Engineering: Compulsory		

Course L2783: Differential Equations 2 (Partial Differential Equations) (EN)	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 2, Study Time in Lecture 28
<b>Lecturer</b>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<p>Main features of the theory and numerical treatment of partial differential equations</p> <ul style="list-style-type: none"> <li>• Examples of partial differential equations</li> <li>• First order quasilinear differential equations</li> <li>• Normal forms of second order differential equations</li> <li>• Harmonic functions and maximum principle</li> <li>• Maximum principle for the heat equation</li> <li>• Wave equation</li> <li>• Liouville's formula</li> <li>• Special functions</li> <li>• Difference methods</li> <li>• Finite elements</li> </ul>
<b>Literature</b>	<a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a>

Course L2784: Differential Equations 2 (Partial Differential Equations) (EN)	
<b>Typ</b>	Recitation Section (large)
<b>Hrs/wk</b>	1
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 16, Study Time in Lecture 14
<b>Lecturer</b>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L2785: Differential Equations 2 (Partial Differential Equations) (EN)	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	1
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 16, Study Time in Lecture 14
<b>Lecturer</b>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L2786: Complex Functions (EN)	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 2, Study Time in Lecture 28
<b>Lecturer</b>	Dozenten des Fachbereiches Mathematik der UHH
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<p>Main features of complex analysis</p> <ul style="list-style-type: none"> <li>• Functions of one complex variable</li> <li>• Complex differentiation</li> <li>• Conformal mappings</li> <li>• Complex integration</li> <li>• Cauchy's integral theorem</li> <li>• Cauchy's integral formula</li> <li>• Taylor and Laurent series expansion</li> <li>• Singularities and residuals</li> <li>• Integral transformations: Fourier and Laplace transformation</li> </ul>
<b>Literature</b>	<a href="http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html">http://www.math.uni-hamburg.de/teaching/export/tuhh/index.html</a>



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

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

Module M1962: Basics space electronics and primary mission			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Basics space electronics and primary mission (L3204)		Project-/problem-based Learning	4
<b>Module Responsible</b>	Prof. Ulf Kulau		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Electrical engineering / Fundamentals of electrical engineering</li> <li>• Computer science / Computer science for engineers</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	<ul style="list-style-type: none"> <li>• Fundamentals of space electronics,</li> <li>• Subcomponents of satellite systems</li> <li>• Fragmentation and planning of primary missions</li> <li>• Active participation in CubeSat mission to apply learned skills</li> <li>• Soft skills in project management, project planning and project communication</li> </ul>		
<i>Skills</i>	<p>Upon completion of the module, students will have learned fundamentals of space electronics. They also know how to plan primary missions and how to define subsystems to achieve this primary mission (requirements analysis, performance specification). They will be actively involved in missions and will be expected to put what they have learned into practice there. Additional soft skills in the area of general project management will be taught and applied through collaboration with the students.</p> <ul style="list-style-type: none"> <li>• Basic teaching</li> <li>• Conceptual design of subsystems (description of requirements and services)</li> <li>• Project planning and fragmentation of primary missions (space missions)</li> <li>• Practical application in CubeSat mission</li> </ul>		
<b>Personal Competence</b>			
<i>Social Competence</i>	The work takes place alternately in the entire group, but also in small groups. This requires close cooperation and coordination within the individual teams. The goal is for students to gain a sound knowledge of space electronics and space missions on the one hand, to apply this knowledge on the other hand and to generate sustainability of their results by working in small groups. This can be, for example, the passing on of the requirement and performance specifications, which act as a basis, starting point and result across semesters.		
<i>Autonomy</i>	After completing the module, students will be able to independently plan and carry out scientific projects and processes. In group work, organization, idea generation, derivation of hypotheses and thought processes are to be independently moderated and carried out.		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written elaboration		
<b>Examination duration and scale</b>	Report on achieved results		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Electrical Engineering: Core Qualification: Elective Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory		

Course L3204: Basics space electronics and primary mission	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	4
<b>CP</b>	6
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56
<b>Lecturer</b>	Prof. Ulf Kulau
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe/SoSe
<b>Content</b>	
<b>Literature</b>	

Module M0651: Computational Geometry			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Computational Geoemetry (L0393)	Lecture	2	4
Computational Geoemetry (L0394)	Recitation Section (small)	2	2
<b>Module Responsible</b>	Dr. Prashant Batra		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<p>Linear algebra and analytic geometry as taught in higher secondary school</p> <p>(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)</p> <p>Basic data structures (trees, binary trees, search trees, balanced binary trees, linked lists)</p> <p>Definition of a graph</p>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> Students can name the basic concepts of computer-assisted geometry, describe them with mathematical precision, and explain them by means of examples.</p> <p>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.</p> <p>Students are able to discuss logical connections between these concepts and to explain them by means of examples.</p> <p><i>Skills</i> 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.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> 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.</p> <p><i>Autonomy</i> Students are capable of accessing independently further logical connections between the concepts about which they have learnt and are able to verify them.</p>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory		

Course L0393: Computational Geometry	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 92, Study Time in Lecture 28
<b>Lecturer</b>	Dr. Prashant Batra
<b>Language</b>	DE
<b>Cycle</b>	WiSe
<b>Content</b>	<p>Construction of the convex hull of <math>n</math> points, triangulation of a simple polygon</p> <p>Construction of Delaunay-triangulation and Voronoi-diagram</p> <p>Algorithms and data structures for the construction of arrangements, and Ham-Sandwich-Cuts.</p> <p>the intersection of half-planes, the optimization of a linear functional over the latter.</p> <p>Efficient determination of all intersection of (orthogonal) lines (line segments)</p> <p>Approximative computation of the diameter of a point set</p> <p>Randomised incremental algorithms</p> <p>Basics of lattice point theory, LLL-algorithm and application in integer-valued optimization.</p> <p>Basics of motion planning</p>
<b>Literature</b>	<p>Computational Geometry Algorithms and Applications Authors:</p> <ul style="list-style-type: none"> <li>• Prof. Dr. Mark de Berg,</li> <li>• Dr. Otfried Cheong,</li> <li>• Dr. Marc van Kreveld,</li> <li>• Prof. Dr. Mark Overmars</li> </ul> <p>Springer e-Book: <a href="http://dx.doi.org/10.1007/978-3-540-77974-2">http://dx.doi.org/10.1007/978-3-540-77974-2</a></p> <p>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: <a href="http://dx.doi.org/10.1007/3-540-27619-X">http://dx.doi.org/10.1007/3-540-27619-X</a></p> <p>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</p> <p>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</p> <p>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)</p>

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

Module M0941: Combinatorial Structures and Algorithms			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b> <b>CP</b>
Combinatorial Structures and Algorithms (L1100)		Lecture	3                  4
Combinatorial Structures and Algorithms (L1101)		Recitation Section (small)	1                  2
<b>Module Responsible</b>	Prof. Anusch Taraz		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>• Mathematics I + II</li> <li>• Discrete Algebraic Structures</li> <li>• Graph Theory and Optimization</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<ul style="list-style-type: none"> <li>• Students can name the basic concepts in Combinatorics and Algorithms. They are able to explain them using appropriate examples.</li> <li>• Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>• They know proof strategies and can reproduce them.</li> </ul> <ul style="list-style-type: none"> <li>• Students can model problems in Combinatorics and Algorithms with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods.</li> <li>• Students are able to discover and verify further logical connections between the concepts studied in the course.</li> <li>• For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.</li> </ul> <ul style="list-style-type: none"> <li>• Students are able to work together in teams. They are capable to use mathematics as a common language.</li> <li>• In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.</li> </ul> <ul style="list-style-type: none"> <li>• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them.</li> <li>• Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>		
<i>Knowledge</i>			
<i>Skills</i>			
<b>Personal Competence</b>			
<i>Social Competence</i>			
<i>Autonomy</i>			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	30 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory		

Course L1100: Combinatorial Structures and Algorithms	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 78, Study Time in Lecture 42
<b>Lecturer</b>	Prof. Anusch Taraz, Dr. Dennis Clemens
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Counting</li> <li>• Structural Graph Theory</li> <li>• Analysis of Algorithms</li> <li>• Extremal Combinatorics</li> <li>• Random discrete structures</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• M. Aigner: Diskrete Mathematik, Vieweg, 6. Aufl., 2006</li> <li>• J. Matoušek &amp; J. Nešetřil: Diskrete Mathematik - Eine Entdeckungsreise, Springer, 2007</li> <li>• A. Steger: Diskrete Strukturen - Band 1: Kombinatorik, Graphentheorie, Algebra, Springer, 2. Aufl. 2007</li> <li>• A. Taraz: Diskrete Mathematik, Birkhäuser, 2012.</li> </ul>

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

Module M2046: Introduction to Quantum Computing				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Introduction to Quantum Computing (L3109)		Lecture	2	3
Introduction to Quantum Computing (L3110)		Recitation Section (large)	2	3
<b>Module Responsible</b>	Prof. Martin Kliesch			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Linear algebra and very good mathematical skills</li> <li>Prior knowledge in theoretical computer science or quantum mechanics is helpful but not required</li> </ul>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <p>Quantum computing is among the most exciting applications of quantum mechanics. Quantum algorithms can efficiently solve computational problems that have a prohibitive runtime on traditional computers. Such problems include, for instance, factoring of integer numbers or energy estimation problems from quantum chemistry and material science.</p> <p>This course provides an introduction to the topic. An emphasis will be put on conceptual and mathematical aspects.</p> <p><i>Skills</i></p> <ul style="list-style-type: none"> <li>Rigorous understanding of how quantum algorithms work and the ability to analyze them</li> <li>Connection of concepts in quantum mechanics and computer science</li> <li>Basic knowledge required to start programming a quantum computer</li> <li>Ability to solve exercises related to quantum algorithms</li> </ul> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p>After completing this module, students are expected to be able to work on subject-specific tasks alone or in a group and to present the results appropriately. Moreover, students will be trained to identify and defuse misleading statements related to quantum computing, which can often be found in popular media.</p> <p><i>Autonomy</i></p> <p>After completion of this module, students are able to work out sub-areas of the subject independently using textbooks and other literature, to summarize and present the acquired knowledge and to link it to the contents of other courses.</p>			
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	No	15 %	Excercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	120 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory General Engineering Science (German program, 7 semester): Specialisation Data Science: Elective Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Engineering Science: Specialisation Data Science: Elective Compulsory Engineering Science: Specialisation Information and Communication Systems: Elective Compulsory Engineering Science: Specialisation Mechatronics: Elective Compulsory Computer Science in Engineering: Specialisation I. Computer Science: Elective Compulsory Technomathematics: Specialisation II. Informatics: Elective Compulsory			

Course L3109: Introduction to Quantum Computing	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Martin Kliesch
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	<p>Quantum computing is among the most exciting applications of quantum mechanics. Quantum algorithms can efficiently solve computational problems that have a prohibitive runtime on traditional computers. Such problems include, for instance, factoring of integer numbers or energy estimation problems from quantum chemistry and material science.</p> <p>This course provides an introduction to the topic. An emphasis will be put on conceptual and mathematical aspects.</p>
<b>Literature</b>	<ul style="list-style-type: none"> <li>Course specific lecture notes will be provided</li> <li>Nielsen and Chuang, Quantum Computation and Quantum Information</li> <li>Sevag Gharibian's lecture notes, Introduction to Quantum Computation</li> </ul>



<b>Course L3110: Introduction to Quantum Computing</b>	
<b>Typ</b>	Recitation Section (large)
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Martin Kliesch
<b>Language</b>	EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1592: Statistics				
<b>Courses</b>				
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Statistics (L2430)		Lecture	3	4
Statistics (L3229)		Project-/problem-based Learning	1	1
Statistics (L2431)		Recitation Section (small)	1	1
<b>Module Responsible</b>	Prof. Matthias Schulte			
<b>Admission Requirements</b>	None			
<b>Recommended Previous Knowledge</b>	Stochastics (or a comparable class)			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results			
<b>Professional Competence</b>	<ul style="list-style-type: none"> <li>• Students can name the basic concepts in Statistics. They are able to explain them using appropriate examples.</li> <li>• Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.</li> <li>• 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.</li> <li>• Students are able to discover and verify further logical connections between the concepts studied in the course.</li> <li>• For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.</li> <li>• 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).</li> <li>• In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers.</li> <li>• Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them.</li> <li>• Students can put their knowledge in relation to the contents of other lectures.</li> <li>• Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.</li> </ul>			
<i>Knowledge</i>				
<i>Skills</i>				
<b>Personal Competence</b>				
<i>Social Competence</i>				
<i>Autonomy</i>				
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70			
<b>Credit points</b>	6			
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b>	<b>Description</b>
	No	10 %	Exercises	
<b>Examination</b>	Written exam			
<b>Examination duration and scale</b>	90 min			
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Advanced Materials: Elective Compulsory General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory General Engineering Science (German program, 7 semester): Specialisation Data Science: Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Core Qualification: Compulsory Engineering Science: Specialisation Advanced Materials: Elective Compulsory Engineering Science: Specialisation Data Science: Compulsory Engineering Science: Specialisation Information and Communication Systems: Compulsory Logistics and Mobility: Specialisation Information Technology: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory Theoretical Mechanical Engineering: Specialisation Robotics and Computer Science: Elective Compulsory Engineering and Management - Major in Logistics and Mobility: Specialisation II. Information Technology: Elective Compulsory			

Course L2430: Statistics	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 78, Study Time in Lecture 42
<b>Lecturer</b>	Prof. Matthias Schulte
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Multivariate distributions and stochastic convergence</li> <li>• Point estimators</li> <li>• Confidence intervals</li> <li>• Hypothesis testing</li> <li>• Nonparametric statistics</li> <li>• Linear Regression</li> <li>• Statistical software (R)</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• L. Dümbgen (2016): Einführung in die Statistik, Birkhäuser.</li> <li>• L. Dümbgen (2003): Stochastik für Informatiker, Springer.</li> <li>• H.-O. Georgii (2012): Stochastics: Introduction to Probability and Statistics, 2nd edition, De Gruyter.</li> <li>• N. Henze (2018): Stochastik für Einsteiger, 12th edition, Springer.</li> <li>• A. Klenke (2014): Probability Theory: A Comprehensive Course, 2nd edition, Springer.</li> <li>• U. Krengel (2005): Einführung in die Wahrscheinlichkeitstheorie und Statistik, 8th edition, Vieweg.</li> </ul>

Course L3229: Statistics	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	1
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 16, Study Time in Lecture 14
<b>Lecturer</b>	Prof. Matthias Schulte
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Course L2431: Statistics	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	1
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 16, Study Time in Lecture 14
<b>Lecturer</b>	Prof. Matthias Schulte
<b>Language</b>	DE/EN
<b>Cycle</b>	WiSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M0715: Solvers for Sparse Linear Systems			
<b>Courses</b>			
<b>Title</b>	<b>Typ</b>	<b>Hrs/wk</b>	<b>CP</b>
Solvers for Sparse Linear Systems (L0583)	Lecture	2	3
Solvers for Sparse Linear Systems (L0584)	Recitation Section (small)	2	3
<b>Module Responsible</b>	Prof. Sabine Le Borne		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	<ul style="list-style-type: none"> <li>Mathematics I + II for Engineering students or Analysis &amp; Lineare Algebra I + II for Technomathematicians</li> <li>Programming experience in C</li> </ul>		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i></p> <p>Students can</p> <ul style="list-style-type: none"> <li>list classical and modern iteration methods and their interrelationships,</li> <li>repeat convergence statements for iterative methods,</li> <li>explain aspects regarding the efficient implementation of iteration methods.</li> </ul> <p><i>Skills</i></p> <p>Students are able to</p> <ul style="list-style-type: none"> <li>analyse, implement, test, and compare iterative methods,</li> <li>analyse the convergence behaviour of iterative methods and, if applicable, compute convergence rates.</li> </ul> <p><b>Personal Competence</b></p> <p><i>Social Competence</i></p> <p>Students are able to</p> <ul style="list-style-type: none"> <li>work together in heterogeneously composed teams (i.e., teams from different study programs and background knowledge), explain theoretical foundations and support each other with practical aspects regarding the implementation of algorithms.</li> </ul> <p><i>Autonomy</i></p> <p>Students are capable</p> <ul style="list-style-type: none"> <li>to assess whether the supporting theoretical and practical excercises are better solved individually or in a team,</li> <li>to work on complex problems over an extended period of time,</li> <li>to assess their individual progress and, if necessary, to ask questions and seek help.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Oral exam		
<b>Examination duration and scale</b>	20 min		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Specialisation I. Mathematics/Computer Science: Elective Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Technomathematics: Specialisation I. Mathematics: Elective Compulsory		

Course L0583: Solvers for Sparse Linear Systems	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sabine Le Borne
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ol style="list-style-type: none"> <li>Sparse systems: Orderings and storage formats, direct solvers</li> <li>Classical methods: basic notions, convergence</li> <li>Projection methods</li> <li>Krylov space methods</li> <li>Preconditioning (e.g. ILU)</li> <li>Multigrid methods</li> <li>Domain Decomposition Methods</li> </ol>
<b>Literature</b>	<ol style="list-style-type: none"> <li>Y. Saad. Iterative methods for sparse linear systems</li> <li>M. Olshanskii, E. Tyrtshnikov. Iterative methods for linear systems: theory and applications</li> </ol>

<b>Course L0584: Solvers for Sparse Linear Systems</b>	
<b>Typ</b>	Recitation Section (small)
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Sabine Le Borne
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

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

Course L0428: Algebra and Control	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 92, Study Time in Lecture 28
<b>Lecturer</b>	Dr. Prashant Batra
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<p>- Algebraic control methods, polynomial and fractional approach</p> <p>- Single input - single output (SISO) control systems synthesis by algebraic methods,</p> <p>- Simultaneous stabilization</p> <p>- Parametrization of all stabilizing controllers</p> <p>- Selected methods of pole assignment.</p> <p>- Filtering and sensitivity minimization</p> <p>- Polynomial matrices, left and right polynomial fractions.</p> <p>- Euclidean algorithm, diophantine equations over rings</p> <p>- Smith-McMillan normal form</p> <p>- Multiple input - multiple output control system synthesis by polynomial methods, condition of stability.</p>
<b>Literature</b>	<ul style="list-style-type: none"> <li>• Vidyasagar, M.: Control system synthesis: a factorization approach. The MIT Press, Cambridge/Mass. - London, 1985.</li> <li>• Vardulakis, A.I.G.: Linear multivariable control. Algebraic analysis and synthesis methods, John Wiley &amp; Sons, Chichester, UK, 1991.</li> <li>• Chen, Chi-Tsong: Analog and digital control system design. Transfer-function, state-space, and algebraic methods. Oxford Univ. Press, 1995.</li> <li>• Kučera, V.: Analysis and Design of Discrete Linear Control Systems. Praha: Academia, 1991.</li> </ul>

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

Module M0634: Introduction into Medical Technology and Systems			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b> <b>CP</b>
Introduction into Medical Technology and Systems (L0342)		Lecture	2                  3
Introduction into Medical Technology and Systems (L0343)		Project Seminar	2                  2
Introduction into Medical Technology and Systems (L1876)		Recitation Section (large)	1                  1
<b>Module Responsible</b>	Prof. Alexander Schlaefer		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	principles of math (algebra, analysis/calculus) principles of stochastics principles of programming, R/Matlab		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	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.		
<i>Skills</i>	The students are able to evaluate systems and medical devices in the context of clinical applications.		
<b>Personal Competence</b>			
<i>Social Competence</i>	The students describe a problem in medical technology as a project, and define tasks that are solved in a joint effort. The students can critically reflect on the results of other groups and make constructive suggestions for improvement.		
<i>Autonomy</i>	The students can assess their level of knowledge and document their work results. They can critically evaluate the results achieved and present them in an appropriate manner.		
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70		
<b>Credit points</b>	6		
<b>Course achievement</b>	<b>Compulsory</b>	<b>Bonus</b>	<b>Form</b> <b>Description</b>
	Yes	10 %	Written elaboration
	Yes	10 %	Presentation
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 minutes		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Biomedical Engineering: Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Specialisation II. Application: Elective Compulsory Electrical Engineering: Core Qualification: Elective Compulsory Engineering Science: Specialisation Biomedical Engineering: Compulsory General Engineering Science (English program, 7 semester): Specialisation Biomedical Engineering: Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory International Management and Engineering: Specialisation II. Medical Engineering: Elective Compulsory International Management and Engineering: Specialisation II. Medical Engineering: Elective Compulsory Mechatronics: Specialisation Medical Engineering: Compulsory Biomedical Engineering: Specialisation Artificial Organs and Regenerative Medicine: Elective Compulsory Biomedical Engineering: Specialisation Implants and Endoprotheses: Elective Compulsory Biomedical Engineering: Specialisation Medical Technology and Control Theory: Elective Compulsory Biomedical Engineering: Specialisation Management and Business Administration: Elective Compulsory Technomathematics: Specialisation III. Engineering Science: Elective Compulsory		



Course L0342: Introduction into Medical Technology and Systems	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	2
<b>CP</b>	3
<b>Workload in Hours</b>	Independent Study Time 62, Study Time in Lecture 28
<b>Lecturer</b>	Prof. Alexander Schlaefer
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>- imaging systems</li> <li>- computer aided surgery</li> <li>- medical sensor systems</li> <li>- medical information systems</li> <li>- regulatory affairs</li> <li>- standard in medical technology</li> </ul> <p>The students will work in groups to apply the methods introduced during the lecture using problem based learning.</p>
<b>Literature</b>	<p>Bernhard Priem, "Visual Computing for Medicine", 2014                      Heinz Handels, "Medizinische Bildverarbeitung", 2009 (<a href="https://katalog.tub.tuhh.de/Record/745558097">https://katalog.tub.tuhh.de/Record/745558097</a>)                      Valery Tuchin, "Tissue Optics - Light Scattering Methods and Instruments for Medical Diagnosis", 2015                      Olaf Drössel, "Biomedizinische Technik - Medizinische Bildgebung", 2014                      H. Gross, "Handbook of Optical Systems", 2008 (<a href="https://katalog.tub.tuhh.de/Record/856571687">https://katalog.tub.tuhh.de/Record/856571687</a>)                      Wolfgang Drexler, "Optical Coherence Tomography", 2008                      Kramme, "Medizintechnik", 2011                      Thorsten M. Buzug, "Computed Tomography", 2008                      Otmar Scherzer, "Handbook of Mathematical Methods in Imaging", 2015                      Weishaupt, "Wie funktioniert MRI?", 2014                      Paul Suetens, "Fundamentals of Medical Imaging", 2009                      Vorlesungsunterlagen</p>

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

Course L1876: Introduction into Medical Technology and Systems	
<b>Typ</b>	Recitation Section (large)
<b>Hrs/wk</b>	1
<b>CP</b>	1
<b>Workload in Hours</b>	Independent Study Time 16, Study Time in Lecture 14
<b>Lecturer</b>	Prof. Alexander Schlaefer
<b>Language</b>	DE
<b>Cycle</b>	SoSe
<b>Content</b>	See interlocking course
<b>Literature</b>	See interlocking course

Module M1269: Lab Cyber-Physical Systems			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Lab Cyber-Physical Systems (L1740)		Project-/problem-based Learning	4
			<b>CP</b>
			6
<b>Module Responsible</b>	Prof. Heiko Falk		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	Module "Embedded Systems"		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>			
<i>Knowledge</i>	Cyber-Physical Systems (CPS) are tightly integrated with their surrounding environment, via sensors, A/D and D/A converters, and actors. Due to their particular application areas, highly specialized sensors, processors and actors are common. Accordingly, there is a large variety of different specification approaches for CPS - in contrast to classical software engineering approaches.		
	Based on practical experiments using robot kits and computers, the basics of specification and modelling of CPS are taught. The lab introduces into the area (basic notions, characteristic properties) and their specification techniques (models of computation, hierarchical automata, data flow models, petri nets, imperative approaches). Since CPS frequently perform control tasks, the lab's experiments will base on simple control applications. The experiments will use state-of-the-art industrial specification tools (MATLAB/Simulink, LabVIEW, NXC) in order to model cyber-physical models that interact with the environment via sensors and actors.		
<i>Skills</i>	After successful attendance of the lab, students are able to develop simple CPS. They understand the interdependencies between a CPS and its surrounding processes which stem from the fact that a CPS interacts with the environment via sensors, A/D converters, digital processors, D/A converters and actors. The lab enables students to compare modelling approaches, to evaluate their advantages and limitations, and to decide which technique to use for a concrete task. They will be able to apply these techniques to practical problems. They obtain first experiences in hardware-related software development, in industry-relevant specification tools and in the area of simple control applications.		
<b>Personal Competence</b>			
<i>Social Competence</i>	Students are able to solve similar problems alone or in a group and to present the results accordingly.		
<i>Autonomy</i>	Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.		
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written elaboration		
<b>Examination duration and scale</b>	Execution and documentation of all lab experiments		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Specialisation Computer Science: Elective Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Computer Science in Engineering: Specialisation II. Mathematics & Engineering Science: Elective Compulsory Mechatronics: Core Qualification: Elective Compulsory		

Course L1740: Lab Cyber-Physical Systems	
<b>Typ</b>	Project-/problem-based Learning
<b>Hrs/wk</b>	4
<b>CP</b>	6
<b>Workload in Hours</b>	Independent Study Time 124, Study Time in Lecture 56
<b>Lecturer</b>	Prof. Heiko Falk
<b>Language</b>	EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>Experiment 1: Programming in NXC</li> <li>Experiment 2: Programming the Robot in Matlab/Simulink</li> <li>Experiment 3: Programming the Robot in LabVIEW</li> </ul>
<b>Literature</b>	<ul style="list-style-type: none"> <li>Peter Marwedel. Embedded System Design - Embedded System Foundations of Cyber-Physical Systems. 2<sup>nd</sup> Edition, Springer, 2012.</li> <li>Begleitende Foliensätze</li> </ul>

Module M0672: Signals and Systems			
<b>Courses</b>			
<b>Title</b>		<b>Typ</b>	<b>Hrs/wk</b>
Signals and Systems (L0432)		Lecture	3
Signals and Systems (L0433)		Recitation Section (small)	2
<b>Module Responsible</b>	Prof. Gerhard Bauch		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>	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.		
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b>	<p><i>Knowledge</i> The students are able to classify and describe signals and linear time-invariant (LTI) systems using methods of signal and system theory. They are able to apply the fundamental transformations of continuous-time and discrete-time signals and systems. They can describe and analyse deterministic signals and systems mathematically in both time and image domain. In particular, they understand the effects in time domain and image domain which are caused by the transition of a continuous-time signal to a discrete-time signal.</p> <p>The students are familiar with the contents of lecture and tutorials. They can explain and apply them to new problems.</p> <p><i>Skills</i> The students are able to describe and analyse deterministic signals and linear time-invariant systems using methods of signal and system theory. They can analyse and design basic systems regarding important properties such as magnitude and phase response, stability, linearity etc.. They can assess the impact of LTI systems on the signal properties in time and frequency domain.</p> <p><b>Personal Competence</b></p> <p><i>Social Competence</i> The students can jointly solve specific problems.</p> <p><i>Autonomy</i> The students are able to acquire relevant information from appropriate literature sources. They can control their level of knowledge during the lecture period by solving tutorial problems, software tools, clicker system.</p>		
<b>Workload in Hours</b>	Independent Study Time 110, Study Time in Lecture 70		
<b>Credit points</b>	6		
<b>Course achievement</b>	None		
<b>Examination</b>	Written exam		
<b>Examination duration and scale</b>	90 min		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Core Qualification: Compulsory Computer Science: Specialisation II. Mathematics and Engineering Science: Elective Compulsory Data Science: Core Qualification: Compulsory Electrical Engineering: Core Qualification: Compulsory Computer Science in 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	
<b>Typ</b>	Lecture
<b>Hrs/wk</b>	3
<b>CP</b>	4
<b>Workload in Hours</b>	Independent Study Time 78, Study Time in Lecture 42
<b>Lecturer</b>	Prof. Gerhard Bauch
<b>Language</b>	DE/EN
<b>Cycle</b>	SoSe
<b>Content</b>	<ul style="list-style-type: none"> <li>• Introduction to signal and system theory</li> <li>• Signals                             <ul style="list-style-type: none"> <li>◦ Classification of signals                                     <ul style="list-style-type: none"> <li>▪ Continuous-time and discrete-time signals</li> <li>▪ Analog and digital signals</li> <li>▪ Deterministic and random signals</li> </ul> </li> <li>◦ Description of LTI systems by differential equations or difference equations, respectively</li> <li>◦ Basic properties of signals and operations on signals</li> <li>◦ Elementary signals</li> <li>◦ Distributions (Generalized Functions)</li> <li>◦ Power and energy of signals</li> <li>◦ Correlation functions of deterministic signals                                     <ul style="list-style-type: none"> <li>▪ Autocorrelation function</li> <li>▪ Crosscorrelation function</li> <li>▪ Orthogonal signals</li> <li>▪ Applications of correlation</li> </ul> </li> </ul> </li> <li>• Linear time-invariant (LTI) systems                             <ul style="list-style-type: none"> <li>◦ Linearity</li> </ul> </li> </ul>

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

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

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

---



---

**Specialization III. Subject Specific Focus**


---



---

**Module M1562: Technical Complementary Course I for CSBS**
**Courses**

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

Module M1568: Technical Complementary Course II for CSBS			
Courses			
Title	Typ	Hrs/wk	CP
<b>Module Responsible</b>	Dozenten des SD E		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b> <i>Knowledge</i> <i>Skills</i> <b>Personal Competence</b> <i>Social Competence</i> <i>Autonomy</i>			
<b>Workload in Hours</b>	Depends on choice of courses		
<b>Credit points</b>	6		
<b>Assignment for the Following Curricula</b>	Computer Science: Specialisation III. Subject Specific Focus: Elective Compulsory		

## Thesis

### Module M1800: Bachelor thesis (dual study program)

#### Courses

Title	Typ	Hrs/wk	CP
<b>Module Responsible</b>	Professoren der TUHH		
<b>Admission Requirements</b>	None		
<b>Recommended Previous Knowledge</b>			
<b>Educational Objectives</b>	After taking part successfully, students have reached the following learning results		
<b>Professional Competence</b> <i>Knowledge</i>	Dual students... <ul style="list-style-type: none"> <li>• ... choose central theoretical principles from their field of study (facts, theories, methods) in relation to problems and applications, present them and discuss them critically.</li> <li>• ... further develop their subject-related and practical knowledge as appropriate and link both areas of knowledge together.</li> <li>• ... present the current research available on a chosen topic or on a chosen operational issue linked to their subject.</li> </ul>		
<b>Professional Competence</b> <i>Skills</i>	Dual students... <ul style="list-style-type: none"> <li>• ... evaluate both the basic knowledge linked to their field of study acquired at the university and professional knowledge gained through the company, then purposefully use it to solve technical and application-related problems.</li> <li>• ... analyse questions and problems using the methods learned throughout their studies (including practical phases), reach factually justifiable decisions and develop application-specific solutions.</li> <li>• ... critically analyse the results of their own research work from a subject-specific and professional perspective.</li> </ul>		
<b>Personal Competence</b> <i>Social Competence</i>	Dual students... <ul style="list-style-type: none"> <li>• ... present a professional problem in the form of an academic question for a specialist audience in a structured, comprehensible and factually correct manner, both orally and in writing.</li> <li>• ... respond to questions as part of a specialist discussion and answer them appropriately. In doing so, they argue their own evaluations and points of view convincingly.</li> </ul>		
<b>Personal Competence</b> <i>Autonomy</i>	Dual students... <ul style="list-style-type: none"> <li>• ... structure a comprehensive, chronological workflow and work independently on a question to a high academic level within a given period of time.</li> <li>• ... identify, develop and link necessary knowledge and material to handle an academic and application-related problem.</li> <li>• ... apply the essential techniques of academic work when conducting their own research on an operational issue.</li> </ul>		
<b>Workload in Hours</b>	Independent Study Time 360, Study Time in Lecture 0		
<b>Credit points</b>	12		
<b>Course achievement</b>	None		
<b>Examination</b>	Thesis		
<b>Examination duration and scale</b>	According to General Regulations		
<b>Assignment for the Following Curricula</b>	General Engineering Science (German program, 7 semester): Thesis: Compulsory Civil- and Environmental Engineering: Thesis: Compulsory Chemical and Bioprocess Engineering: Thesis: Compulsory Computer Science: Thesis: Compulsory Data Science: Thesis: Compulsory Electrical Engineering: Thesis: Compulsory Engineering Science: Thesis: Compulsory Green Technologies: Energy, Water, Climate: Thesis: Compulsory Computer Science in Engineering: Thesis: Compulsory Mechanical Engineering: Thesis: Compulsory Mechatronics: Thesis: Compulsory Naval Architecture: Thesis: Compulsory Technomathematics: Thesis: Compulsory Engineering and Management - Major in Logistics and Mobility: Thesis: Compulsory		