Aschaffenburg University of Applied Sciences
Faculty of Engineering and Computer Science

Module Handbook

for the Bachelor Study Programme

Software Design International

Issued for the degree programme "Software Design International" of the Aschaffenburg University of Applied Sciences by decision of the Dean on 26.08.2026 and by resolution of the Faculty Council of the Faculty of Engineering and Computer Science on xx.xx.2026.

This Module Handbook applies in conjunction with the Study and Examination Regulations dated 01.08.2023 (SPO11), amended by status dated 15.07.2025.

Prof. Dr. Vaupel, Dekan (Dean)

Status: 26.08.2026

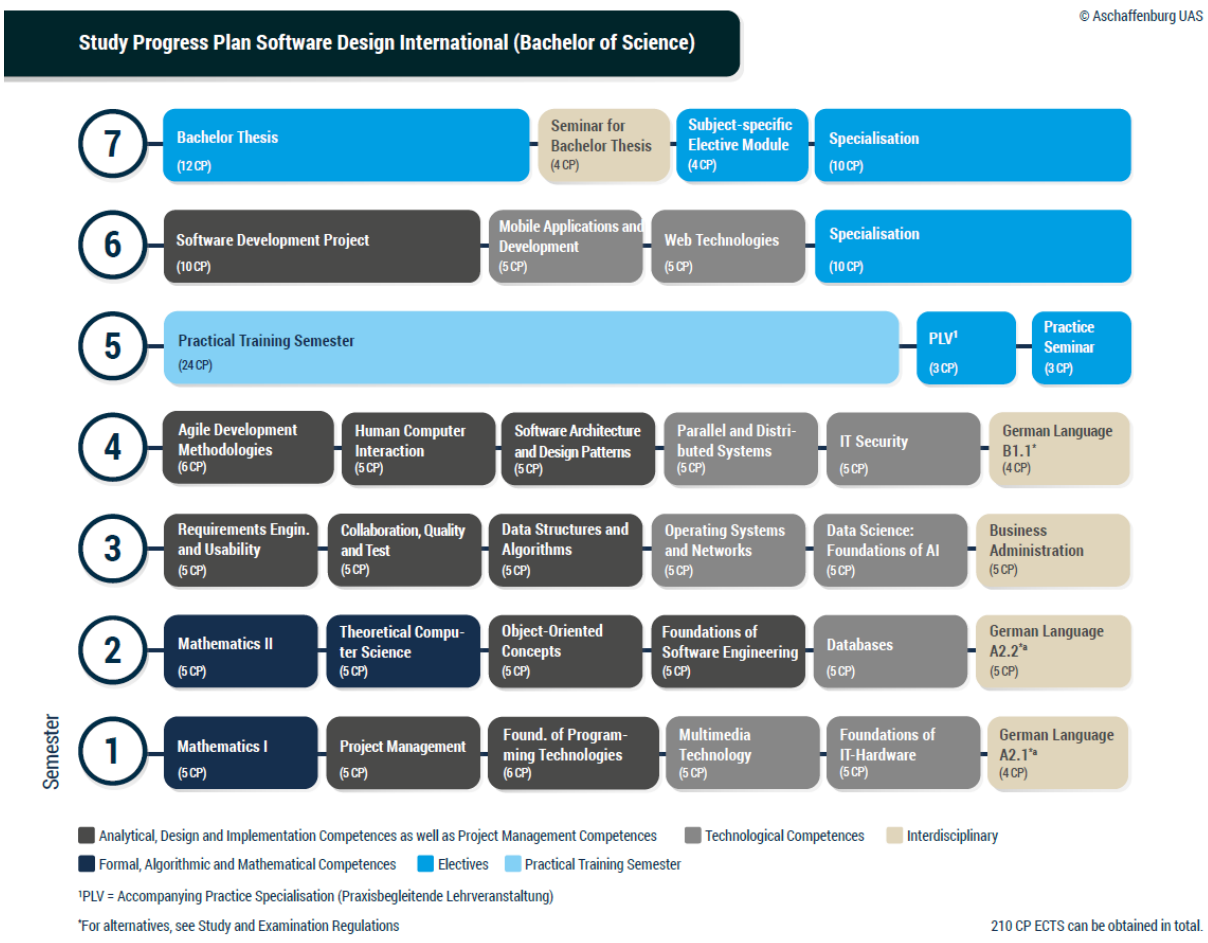
For further information on the modules, the subjects and the respective examinations and certificates of achievement, please refer to the Study and Examination Regulations (Studien- und Prüfungsordnung, SPO) and the study plan of your degree program as amended from time to time.

Study Programme Software Design International

This module handbook is intended to help you understand the academic structure of the Software Design International (B.Sc.) programme and to plan your studies in a well-informed way. It provides detailed descriptions of all modules, including learning outcomes, course content, teaching and assessment formats, and the expected workload (ECTS/total hours). Where relevant, it also outlines recommended prior knowledge, prerequisites, and how each module fits into the overall curriculum.

The programme is structured in clearly consecutive study phases that take you from the fundamentals to independent final work. In the early semesters, you acquire core competencies in programming, mathematics, and fundamental computer science, providing the basis for more advanced content. This is followed by modules that qualify you along the software lifecycle—from requirements and specification through design and software architecture to databases and networks, quality assurance and testing, and IT security. Practice-oriented teaching formats and project work accompany this development throughout. A key milestone is the practical training semester, where you apply and reflect on your knowledge in a professional environment. In the later semesters, you further develop your profile through electives and a specialisation track before completing the programme with the software development project and the bachelor's dissertation. Even as AI-assisted programming becomes more prevalent, requirements, specification, teamwork, and agile collaboration remain central; the curriculum is structured to develop and strengthen these skills step by step.

Use this handbook both as a reference and as a planning tool: compare module outcomes and contents, take note of recommended sequences and prerequisites, and select electives in line with your interests and career goals.



Content

Module: SDI_01, Mathematics I 4
Module: SDI_02, Mathematics II 5
Module: SDI_03, Foundations of Programming Technologies 6
Module: SDI_04, Object-oriented Concepts 7
Module: SDI_05, Data Structures and Algorithms 8
Module: SDI_06, Project Management 9
Module: SDI_07, Foundations of IT-Hardware 10
Module: SDI_08, Multimedia Technology 11
Module: SDI_09, Theoretical Computer Science 12
Module: SDI_10, Foundations of Software Engineering 13
Module: SDI_11, Databases 14
Module: SDI_12, Requirements Engineering and Usability 15
Module: SDI_13, Collaboration, Quality and Test 16
Module: SDI_14, Operating Systems and Networks 17
Module: SDI_15, Data Science: Foundations of AI 19
Module: SDI_16, Agile Development Methodologies 20
Module: SDI_17, Human Computer Interaction 21
Module: SDI_18, IT Security 22
Module: SDI_19, Software Architecture and Design Patterns 23
Module: SDI_20, Parallel and Distributed Systems 24
Module: SDI_21, German as foreign Language A2 25
Module: SDI_22, German as Foreign Language B1 26
Module: SDI_23, Business Administration 27
Module: SDI_25, Mobile Applications and Development 28
Module: SDI_26, Web Technologies 29
Module: SDI_27, Software Development Project 30
Module: SDI_28, Subject-specific compulsory Elective Module 31
Module: SDI_29, Practical Training Semester 32
Module: SDI_30, Practical Course 33
Module: SDI_31, Practical Seminar 34
Module: SDI_32, Seminar for Bachelor Thesis 35
Module: SDI_33, Bachelor Thesis 36
Abbreviations 37

Module: SDI_01, Mathematics I

Module Title	Mathematics I
Module Code	SDI_01
Course(s)	LV1: SDI_01a Mathematics I (seminar course) LV2: SDI_01b Practice for Mathematics I (practice)
Lecturer(s)	Prof. Dr. Klaus Radke, Prof. Dr. Barbara Sprick
Module coordinator(s)	Prof. Dr. Klaus Radke, Prof. Dr. Galia Weidl
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 1, Winter semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 15h preparation, 50h follow-up, 25h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS practice)
ECTS credits	5
Participation prerequisites	None
Applicability of the module	The module teaches the basic mathematical concepts and procedures required in applied computer science. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	Discrete Structures and Linear Algebra Students … • formulate and analyze statements in propositional logic and first-order logic and translate simple problems into formal models. • apply fundamental proof techniques (direct proof, proof by contradiction, mathematical induction) in basic contexts. • use sets, relations, important elementary functions and functions confidently and examine their key properties. • apply concepts from elementary number theory and combinatorics to solve discrete problems. • perform calculations with vectors and matrices and solve systems of linear equations using standard methods. • describe basic properties of vector spaces and linear mappings using simple examples. • model application problems using graphs and apply elementary graph-theoretic methods in basic contexts. • recognize discrete mathematical structures in computer science–related problems and present solutions in a formally correct manner.
Contents	LV1: Discrete Structures and Linear Algebra: • Propositional Logic and First-Order Logic • Sets, Relations, Functions • Important elementary functions • Proof Methods • Elementary Number Theory • Elementary Combinatorics • Vector Spaces, Matrices, Systems of Equations • Graph Theory LV2: • Practice for the contents in LV1
Study / Examination method	See document „Study and examination regulations (SPO)" Possible bonus points: processing of exercises with presentation
Types of media employed	Blackboard, projector, transparencies
Reading list	• Lehman, Leighton, Meyer: Mathematics for Computer Science • Levin: Discrete Mathematics – An Open Introduction • Rosen: Discrete Mathematics and its Applications, Mc Graw Hill (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_02, Mathematics II

Module Title	Mathematics II
Module Code	SDI_02
Course(s)	LV1: SDI_02a Mathematics II (seminar course) LV2: SDI_02b Practice for Mathematics II (practice)
Lecturer(s)	Prof. Dr. Klaus Radke
Module coordinator(s)	Prof. Dr. Klaus Radke, Prof. Dr. Galia Weidl
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 2, Summer semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 15h preparation, 50h follow-up, 25h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS practice)
ECTS credits	5
Participation prerequisites	Recommended: SDI_01 Mathematics I
Applicability of the module	The module teaches the basic mathematical concepts and procedures required in applied computer science. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	Students … • explain the fundamental concepts of analysis (convergence, continuity, differentiability, integrability) and apply them to typical examples. • compute derivatives and integrals of single-variable functions and interpret the results in applied contexts. • analyze and compute partial derivatives of multivariate functions and use them to solve basic optimization problems. • model simple application problems using methods of differential and integral calculus. • explain the concept of probability, including conditional probability, and apply it to discrete problem settings. • compute and interpret basic statistical measures (e.g., mean, variance, standard deviation) for given data sets. • describe and apply fundamental discrete and continuous probability distributions in simple contexts. • perform basic parameter estimation procedures and interpret the results appropriately.
Contents	LV1: Analysis: • Convergence, Continuity, Differentiability, Integrability • Differential and Integral Calculus • Multivariable Functions Probability Theory and Statistics: • Probability and Conditional Probability • Statistical Measures • Probability Distributions • Parameter Estimation LV2: • Practice for the contents in LV1
Study / Examination method	See document „Study and examination regulations (SPO)“ Possible bonus points: processing of exercises with presentation
Types of media employed	Blackboard, projector, transparencies
Reading list	• Hughes-Hallett, Gleason, McCallum: Calculus: Single and Multivariable, Wiley • Diez, Barr, Cetinkaya-Rundel: OpenIntro Statistics (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_03, Foundations of Programming Technologies

Module Title	Foundations of Programming Technologies
Module Code	SDI_03
Course(s)	LV1: SDI_03a Foundations of Programming Technologies (seminar course) LV2: SDI_03b Practice for Foundations of Programming Technologies (practice / project)
Lecturer(s)	Prof. Dr. Martin Sträßer
Module coordinator(s)	Prof. Dr. Alison McNamara
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 1, Winter semester
Workload	Total effort: 180 h (of which: Contact hours: 60h, Independent study: 120h (of which: 40h preparation, 40h follow-up, 40h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS exercise/practices)
ECTS credits	6
Participation prerequisites	None
Applicability of the module	The module teaches the basic concepts and procedures of programming that are required in applied computer science. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • explain the significance of computer science in the context of digitalization and digital change • explain basic concepts of programming languages such as syntax, semantics, data types, memory structures, method calls and parameter passing using a programming language as an example. • write syntactically and semantically correct programs according to a given algorithm in the learned programming language. • explain essential development steps for the Java language to executable programs. • Perform the development steps in an environment (IDE) for the programming language. • implement tasks in their own programs in the programming language. • implement a given simple search or sorting algorithm in the programming language. • explain basic object-oriented concepts, such as classes, inheritance, and polymorphism. • implement tasks in their own programs in the given programming language.
Contents	LV1: • Basics of computer science (history, importance of computer science in the digital age, areas of computer science). • Encoding of information (ASCII, Unicode), number systems (dual, octal, decimal, hexadecimal) • Introduction to programming using Java as an example • Syntax and semantics of the Java programming language, control structures, simple data types and reference data types. • Basic principles of object orientation (generalization, inheritance, encapsulation, polymorphism) • Simple data structures and algorithms LV2: • Introduction to the Eclipse development environment for programming • Creation of programs in above mentioned development environment in above mentioned programming language • Intensive practical application of all concepts presented within the framework of many hands-on exercises
Study / Examination method	See document “Study and examination regulations (SPO)”
	Possible bonus points: completion of computer lab exercises and a presentation
Types of media employed	Blackboard, projector, demonstration, sample programs, worksheets
Reading list	• Bates, K. (2021). Head First Java, 3rd Edition. O'Reilly Media, Inc, USA. • Deitel, P.J. and Deitel, H.M. (2018). Java: how to program: early objects. New York, Ny Pearson. (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_04, Object-oriented Concepts

Module Title	Object-oriented Concepts
Module Code	SDI_04
Course(s)	LV1: SDI_04a Object-oriented Concepts (seminar course) LV2: SDI_04b Practice for Object-oriented Concepts (practice / project)
Lecturer(s)	Prof. Dr. Alison McNamara
Module coordinator(s)	Prof. Dr. Alison McNamara
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 2, Summer semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 30h follow-up, 30h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar-based teaching, LV2: 2 SWS practice/internship)
ECTS credits	5
Participation prerequisites	SDI_03, Foundations of Programming Technologies
Applicability of the module	The module teaches in-depth programming methods that are required in applied computer science. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • outline programming paradigms such as imperative, object-oriented, and declarative programming and explain them using language constructs. • evaluate the suitability of different programming paradigms and programming languages for various application tasks. • explain advanced object-oriented concepts such as interfaces, abstract classes, dynamic binding using the Java programming language. • write syntactically and semantically correct OO programs in the Java language according to a given algorithm and test it. • independently implement generic types and collections in their own Java programs. • map UML diagrams to matching elements in Java. • write simple unit tests and develop test-driven. • perform code reviews, apply "clean code" principles and engage in pair programming.
Contents	LV1: • Programming languages and concepts • Introduction to advanced concepts of object-oriented programming using Java as an example (abstract classes, interfaces, collections, exception handling, etc.) • Deployment • Software testing, test-driven development, unit tests • Clean Code Principles LV2: • Software technical solution of a task from the requirement to the implementation (incl. test) • Practical application of all concepts in the context of assignments • Team practice for the contents of LV 1
Study / Examination method	See document “Study and examination regulations (SPO)” Possible bonus points: completion of computer lab exercises and a presentation
Types of media employed	Blackboard, projector, demonstration, sample programs, worksheets
Reading list	• Bates, K. (2021). Head First Java, 3rd Edition. O'Reilly Media, Inc, USA. • Deitel, P.J. and Deitel, H.M. (2018). Java: how to program: early objects. New York, Ny Pearson. (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_05, Data Structures and Algorithms

Module Title	Data Structures and Algorithms
Module Code	SDI_05
Course(s)	LV1: SDI_05a Data Structures and Algorithms (seminar course) LV2: SDI_05b Practice for Data Structures and Algorithms (practice / project)
Lecturer(s)	Prof. Dr. Barbara Sprick
Module coordinator(s)	Prof. Dr. Barbara Sprick
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 3, Winter semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 30h follow-up, 30h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS exercise/practices)
ECTS credits	5
Participation prerequisites	Recommended: SDI_03, Foundations of Programming Technologies, SDI_04 Object-oriented Concepts
Applicability of the module	The module teaches in-depth algorithms and data structures required in applied computer science. The contents and application examples are essentially aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • can explain the most important basic data structures and algorithms in computer science • can determine the runtime behavior and memory requirements of algorithms and compare the efficiency of algorithms • can explain basic data types (lists, stacks, queues, trees) and implement them in Java. • can explain the influence of a chosen data structure on the realization of an algorithm • know selected algorithm design principles and can explain and apply them in smaller examples • know strategies for selecting algorithms and can identify suitable algorithms for selected standard problems in computer science and use them to solve the problem
Contents	LV1: • Big O-notation, complexity of algorithms • Basic data structures (array, stack, queue, lists, ...) • algorithms for searching and sorting • Data structures Hashes, Trees and Dictionaries • Graphs and graph algorithms LV2: • Team exercises and PBL on the contents of LV 1. • Implementation of the presented algorithms
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points: none
Types of media employed	Blackboard, projector, demonstration, sample programs, worksheets
Reading list	• Thomas H. Cormen, Charles E. Leiserson et al.: Introduction to Algorithms, The MIT Press • Robert Sedgewick, Kevin Wayne: Algorithms, 4th edition, Pearson Studium (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_06, Project Management

Module Title	Project Management
Module Code	SDI_06
Course(s)	LV1: SDI_06a Project Management (seminar course) LV2: SDI_06b Project for Project Management (practice / project)
Lecturer(s)	Prof. Dr. Timea Illes-Seifert
Module coordinator(s)	Prof. Dr. Timea Illes-Seifert
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 1, Winter semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 30h follow-up, 30h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS exercise)
ECTS credits	5
Participation prerequisites	None
Applicability of the module	The module teaches the basic concepts and procedures of project management that are required in applied computer science. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • Know the essential characteristics of a project and understand the difference between a project and a routine activity. • describe the product vision in a product vision board and apply the SMART template for goal formulation. • discuss the difference between classic and agile project planning and control. • identify subtasks and summarize them in a work breakdown structure. • identify stakeholders, recognize their influence on project success and develop measures to manage them. • identify risks and propose measures to reduce or avoid them. • apply different creativity techniques to develop ideas or solve problems and present their (intermediate) results. • plan and control a sample project in a team and apply the methods learned. • Apply different methods for retrospective.
Contents	• Introduction project management: terms, standards, characteristics of projects, project success • Agile vs. classical project management, process models, SCRUM • Project initialization and definition: mission statement, kick-off and initial brainstorming, product vision, project environment, stakeholder analysis, project goals, risk management, requirements analysis • Project planning: project management phases, phase plan, milestone planning, work breakdown structure, effort estimation • Project control: stages of project controlling, project status and progress, milestone and cost trend analysis • Project closure: documentation, lessons learned • Conflict management, teamwork, feedback, creativity techniques • Agile project management using the example of SCRUM
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points for LV1: none Possible bonus points for LV2: none
Types of media employed	Blackboard, beamer, demonstration, laboratory experiments
Reading list	• Jürg Kuster, Eugen et al.: Project Management Handbook, Springer • Olson, David Louis: Information systems project management, Business Expert Press (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_07, Foundations of IT-Hardware

Module Title	Foundations of IT Hardware
Module Code	SDI_07
Course(s)	LV1: SDI_07a: Foundations of IT-Hardware (seminar course) LV2: SDI_07b: Practice for Foundations of IT-Hardware (practice)
Lecturer(s)	Prof. Dr. Jóakim von Kistowski
Module coordinator(s)	Prof. Dr. Jóakim von Kistowski
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 1, Winter semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 30h follow-up, 30h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS practice)
ECTS credits	5
Participation prerequisites	None
Applicability of the module	The module teaches the basic concepts and methods of IT hardware required in applied computer science. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • explain and designate basic components of a computer. • classify performance characteristics of computers and processors. • understand and characterize IT hardware in terms of its internal structure (throughput, latency, parallelizability, ...). • represent components of a hardware circuit. • perform translation steps of a program written in high-level language into machine code. • set up their own machine program. • reproduce the execution of machine code on a processor. • Outline the communication between hardware and software components. • use Boolean algebra to describe and optimize hardware circuits.
Contents	• Components, functioning and structure of computer systems • Microprocessor and memory structure • Basic building blocks of digital technology • Design of hardware components by means of hardware description languages • Memory organization and hierarchy in the computer • Connection of software creation and execution of the software on the hardware • Interaction of computer components in heterogeneous architectures (embedded systems with actuators/sensors, client-server architectures, ...) • Optimization strategies for hardware: caching, pipelining, parallelization LV1: • Acquisition of the above-mentioned knowledge in seminar classes LV 2: • Practical application of concepts developed in LV1 using worksheets and calculator exercises.
Study / Examination method	See document „Study and examination regulations (SPO)“ Possible bonus points for LV1: none Possible bonus points for LV2: none
Types of media employed	Blackboard, beamer, worksheets, working on the computer
Reading list	• Patterson, Hennessy: Computer Organization and Design: The Hardware/Software Interface, Elsevier Science • Harris, Harris: Digital Design and Computer Architecture, Morgan Kaufmann (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_08, Multimedia Technology

Module Title	Multimedia Technology
Module Code	SDI_08
Course(s)	LV1: SDI_08a: Multimedia Technology LV2: SDI_08b: Practice for Multimedia Technology
Lecturer(s)	Prof. Dr. Alison McNamara
Module coordinator(s)	Prof. Dr. Alison McNamara
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 1, Winter semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 30h follow-up, 30h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS practice)
ECTS credits	5
Participation prerequisites	None
Applicability of the module	The module teaches the basic concepts and procedures of multimedia technology as they are required in applied computer science. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • define the term multimedia and name different types of digital media and their specifics. • describe the process of digitization and its impact on the recording and optimization of digital media. • refer and evaluate different image, sound and video formats and the codecs and compression methods used in their creation. • explain available tools and workflows for conceptualizing, designing, and creating multimedia content and applications. • Apply basic principles of user centred design to the process of implementing web applications and mobile apps: from conception to ingestion, preparation, publication, and evaluation. • use agile strategies as well as creative tools to design multimedia software applications for different target platforms (design thinking and rapid prototyping using scribbles, wireframes, click dummies). • convincingly present their own concepts and implementations and self-critically discuss the strengths and weaknesses of design solutions. • systematically analyse and discuss their own and other people's design solutions with regard to the user's requirements for multimedia applications and the human-machine interface. • master the challenges of developing design solutions in project teams through self-reflection and active communication within initial application projects.
Contents	LV1: • Definition of multimedia and overview of types of digital media • Overview of the process of digitization: The path from analogue to digital image and sound signals • Overview of digital image, sound, and video formats and the role and function of compression and codecs. • Importance of input and output devices for multimedia applications (controllers, touch input, gestures, display technologies) • User-centred use of multimedia content in web applications and mobile apps: Process of design/recording, preparation, publication and evaluation. • Agile strategies, tools and steps in user-centred design of software applications for different target platforms: Design thinking, scribbles, wireframes, rapid prototyping. LV2: • User-centred and agile planning, conception, design, implementation and evaluation of multimedia applications for different target platforms • Critical analysis and evaluation of the implementations developed during the exercises.
Study / Examination method	See document „Study and examination regulations (SPO)“ Possible bonus points: completion of computer lab exercises and a presentation
Types of media employed	Blackboard, beamer, worksheets, digital teaching content
Reading list	• Peter Bühler et al.: Bibliothek der Mediengestaltung, Springer Vieweg Verlag, Bände „Digitales Bild“, „AV-Medien“, „Digital Publishing“, „Präsentation“ (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_09, Theoretical Computer Science

Module Title	Theoretical Computer Science
Module Code	SDI_09
Course(s)	LV1: SDI_09a: Theoretical Computer Science LV2: SDI_09b: Practice for Theoretical Computer Science
Lecturer(s)	Prof. Dr. Barbara Sprick
Module coordinator(s)	Prof. Dr. Barbara Sprick
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 2, Summer semester
Workload	Total effort: 150h (thereof Contact hours: 60h, Independent study: 90h (thereof: 30 preparation, 30h follow-up, 30h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS practice)
ECTS credits	5
Participation prerequisites	None
Applicability of the module	The module teaches the theoretical foundations of computer science that are required in applied computer science. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students... • describe and apply formal languages. • describe the basics of automata theory and analyse given automata. • describe Turing machines and apply them to given problem statements. • describe and apply regular languages. • apply the basic principles of regular expressions to given problem statements. • describe and apply grammars. • name runtime and complexity classes. • recognise runtime differences of different algorithms. • make statements about the computability and decidability of problems.
Contents	• Formal languages • Automata theory and Turing machines • Regular languages • Grammars • Computability and decidability • Complexity theory and complexity/runtime classes LV1: Elaboration of the above-mentioned contents in seminar-based instruction. LV2: Practical practice of the contents taught in LV1 by means of exercises.
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points for LV1: none Possible bonus points for LV2: none
Types of media employed	Blackboard, projector, worksheets
Reading list	• Hopcroft, Motwani, Ullmann: Introduction to Automata Theory, Languages and Computation • Michael Sipser: Introduction to the Theory of Computation (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_10, Foundations of Software Engineering

Module Title	Foundations of Software Engineering
Module Code	SDI_10
Course(s)	LV1: SDI_10a: Foundations of Software Engineering (seminar course) LV2: SDI_10b: Project for Foundations of Software Engineering (practice / project)
Lecturer(s)	Prof. Dr. Jóakim von Kistowski
Module coordinator(s)	Prof. Dr. Timea Illes-Seifert
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 2, Summer semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 35h follow-up, 25h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar-based teaching, LV2: 2 SWS project/exercise)
ECTS credits	5
Participation prerequisites	Recommended: SDI_03 Foundations of Programming Technologies, SDI_06 Project Management
Applicability of the module	The module teaches the basic concepts and procedures of software engineering that are required in applied computer science. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • characterize traditional process models (waterfall, V-model, V-model XT) and agile (Scrum, XP) process models. • plan solve a development problem using a process model and perform software development tasks in teams. • explain and apply methods, languages and tools for the individual disciplines: requirements specification, architecture and design, implementation and testing. • draw conclusions about the effectiveness of the methods applied. • apply UML models. • apply object-oriented analysis and design to concrete examples. • show why quality assurance is a permanent task in software development and explain different methods of analytical and constructive quality assurance. • moderate reviews and provide constructive feedback on the "test object". • differentiate between preventive, perfective and corrective maintenance and explain why test automation is essential for efficient maintenance. • Conduct team retrospectives.
Contents	LV1: • Basics of software engineering, basic terms, modelling • Process models, evaluation and improvement of the software process, agile software development • Software engineering disciplines: requirements engineering, architecture and design, implementation, testing • Software management: quality management, software project management, configuration management, the human being in software development, metrics • Reuse, software maintenance, reengineering LV2: • Team practice for the contents of LV 1
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points for LV1: none Possible bonus points for LV2: none
Types of media employed	blackboard, beamer, demonstration
Reading list	• Martin Fowler: Refactoring: Improving the Design of Existing Code, Pearson Addison-Wesley Signature Series • Ian Sommerville: Engineering Software Products: An Introduction to Modern Software Engineering, Pearson • Ian Sommerville: Software Engineering, Pearson • Andreas Spillner, Tilo Linz: Software Testing Foundations: A Study Guide for the Certified Tester Exam- Foundation Level- ISTQB® Compliant, dpunkt.verlag GmbH • Unhelkar, Bhuvan: Software engineering with UML, CRC Press • Martin Fowler: Refactoring: Improving the Design of Existing Code, Pearson Addison-Wesley Signature Series (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_11, Databases

Module Title	Databases
Module Code	SDI_11
Course(s)	LV1: SDI_11a: Databases (seminar course) LV2: SDI_11b Practice for Databases (practice)
Lecturer(s)	Prof. Dr. Barbara Sprick
Module coordinator(s)	Prof. Dr. Barbara Sprick
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 2, Summer semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 35h follow-up, 25h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS practice)
ECTS credits	5
Participation prerequisites	None
Applicability of the module	The module teaches the basic concepts and procedures of databases that are required in applied computer science. The contents and application examples are essentially aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • can explain basic concepts and theoretical foundations of relational database systems and relational query languages using technical terms • can distinguish different database models from each other • can explain the basic structure of database management systems using technical terms • discuss privacy mechanisms and societal implications of big data collection. • explain different types of knowledge and distinguish knowledge processing from databases. • Can design conceptual database blueprints and translate them into normalized relational database schemas • can perform complex SQL queries and SQL change operations • can create small database applications • estimate the value of a designed database schema for a • Users one.
Contents	LV1: • Basic concepts of relational databases • Relational query languages, especially SQL • Database Modeling • Database Management Systems • Normalized relational database schemas • Multi-user operation of databases, transaction management, scheduling • Assignment of rights in SQL, access control models • NoSQL LV2: • Practice for the contents of LV 1
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points for LV1: none Possible bonus points for LV2: none
Types of media employed	Blackboard, projector, demonstration
Reading list	• Elmasri, Navathe: Fundamentals of Databases, Global Edition • Garcia-Molina: Database Systems: The Complete Book (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_12, Requirements Engineering and Usability

Module Title	Requirements Engineering and Usability
Module Code	SDI_12
Course(s)	LV1: SDI_12a: Requirements Engineering and Usability (seminar course) LV2: SDI_12b: Practice for Requirements Engineering and Usability (practice)
Lecturer(s)	N.N.
Module coordinator(s)	Prof. Dr. Barbara Sprick
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 3, Winter semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 35h follow-up, 25h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS exercise)
ECTS credits	5
Participation prerequisites	Recommended: SDI_10 Fundamentals of Software Engineering
Applicability of the module	The module teaches basic terms, concepts and procedures of requirements engineering and usability engineering that are required and used in software development. The contents and application examples are essentially aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • define and explain core terms and terminology from the field of requirements engineering, requirements management and usability engineering. • name, explain, and distinguish techniques and methods for eliciting, reviewing, tuning, detailing, managing, and documenting requirements for software systems. • independently select and apply suitable requirements engineering techniques, methods and tools for given project situations. • apply the human-centered design process in the lifecycle of software projects in a planned manner to optimize usability, user experience, and accessibility. • Interpret, survey, and evaluate usage context and requirements, effectiveness, efficiency, and satisfaction for software products.
Contents	LV1: Requirements Engineering and Usability • Basics of requirements engineering and enterprise modeling • Techniques of requirements elicitation and requirements documentation • Review and coordination of requirements in software projects, methods and tools for managing requirements (requirements management) • Definition and meaning of usability, user experience (UX) and accessibility • The human-centered design process according to DIN EN ISO 9241-210 • Target group analysis, usage requirements and usage context • Usability testing: methods and use in the development process and product life cycle • Usability control loop and quality assurance LV 2: Practice for Requirements Engineering and Usability • Team practice for the contents of LV 1 • Exercise of the contents from LV1 on the basis of an own project • Use of design thinking and software tools
Study / Examination method	See document „Study and examination regulations (SPO)“ Possible bonus points for LV1: none Possible bonus points for LV2: none
Types of media employed	Blackboard, beamer, demonstration, digital teaching content, working on the computer
Reading list	• Klaus Pohl: Requirements Engineering. Fundamentals, Principles and Techniques, Springer Verlag • Klaus Pohl, Chris Rupp: Requirements Engineering Fundamental: A Study Guide for the Certified Professional for Requirements Engineering Exam – Foundation Level – IREB compliant, Rocky Nook • Suzanne Robertson, James Robertson: Mastering the Requirements Process: Getting Requirements Right, Addison-Wesley Professional (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_13, Collaboration, Quality and Test

Module Title	Collaboration, Quality and Test
Module Code	SDI_13
Course(s)	LV1: SDI_13a Collaboration, Quality and Test (seminar course) LV2: SDI_13b Practice for Collaboration, Quality and Test (practice / project)
Lecturer(s)	N.N.
Module coordinator(s)	Prof. Dr. Timea Illes-Seifert
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 3, Winter semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 35h follow-up, 25h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS practical course/exercise)
ECTS credits	5
Participation prerequisites	Recommended: SDI_03 Foundations of Programming Technologies, SDI_04 Object-oriented Concepts, SDI_05 Data Structures and Algorithms, SDI_06 Project Management, SDI_10 Foundations of Software Engineering
Applicability of the module	The module teaches the basic concepts and procedures from the areas of collaboration, quality management and testing that are required in applied computer science. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • explain core testing and quality assurance terms and summarize quality management tasks. • explain the characteristics of different quality management systems. • explain the relationships between software development and testing activities and describe typical objectives of testing. • explain and apply different black-box, white-box and experience-based testing procedures. • explain the difference between static and dynamic methods. • explain the purpose of confirmation and regression testing. • Summarize review process activities and apply a review process to a work product. • summarize the purpose and content of the test plan and identify factors that influence the testing effort. • independently select and apply suitable quality assurance techniques, methods and tools for given project situations. • recall, compare and apply methods and tools of collaborative software development.
Contents	LV1: Collaboration, Quality and Test • Basics of quality management (costs/benefits, roles, norms and standards, learning processes). • Quality models and measurement of software quality • Basics of testing (terms, testing process, psychology of testing). • Testing in the software development cycle (test levels, test types, maintenance testing) • Static test (basics, review process) • Test procedure (black-box, white-box, experience-based) • Test of non-functional properties • Test management (test organization, test planning, test control) risk management, defect management, tool support) • Agile testing LV 2: Practice for Collaboration, Quality and Test • Practice for teaching content from LV1 • Optional and voluntary certification to Certified Tester Foundation Level
Study / Examination method	Written See document „Study and examination regulations (SPO)“
	Possible bonus points for LV1: none Possible bonus points for LV2: none
Types of media employed	blackboard, beamer, demonstration, practical work
Reading list	• Andreas Spillner, Tilo Linz: Software Testing Foundations: A Study Guide for the Certified Tester Exam- Foundation Level- ISTQB® Compliant, dpunkt.verlag GmbH • Glenford Myers: The Art of Software Testing, John Wiley & Sons, New York NY (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_14, Operating Systems and Networks

Module Title	Operating Systems and Networks I
Module Code	SDI_14
Course(s)	LV1: SDI_14a Operating Systems and Networks (seminar course) LV2: SDI_14b Practice for Operating Systems and Networks (practice / project)
Lecturer(s)	Prof. Dr.-Ing. Jörg Abke
Module coordinator(s)	Prof. Dr.-Ing. Jörg Abke
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 3, Winter semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 35h follow-up, 25h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS exercise/practices)
ECTS credits	5
Participation prerequisites	Recommended: SDI_03 Foundations of Programming Technologies, SDI_04 Object-oriented Concepts
Applicability of the module	The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students will ... • describe basic concepts and theoretical foundations of operating systems using technical terms. • describe operating system specific programs (e.g., in a shell) and explain their function. • explain approaches of virtualisation with technical terms. • evaluate basic concepts regarding safety and security • operate and use operating system specific programs. • explain basic programming interfaces (API) to an operating system, they list their syntax and address their functions using a higher programming language. • explain the tasks and functions of the layers of the ISO/OSI model and assign the most important service representatives to the respective layer. • describe how the Internet Protocol works at its core and in the end systems. They will assign the network components to the layers. • reproduce the concepts of the protocols (e.g., TCP, IP, UDP) and describe how they work, e.g., using message sequence diagrams. • describe and use system-specific network utilities and their use and apply them to given problems. • explain the requirements for important protocols, e.g., for synchronization and mutual exclusion, for consistency and replication of data, for fault tolerance and for security. • evaluate network protocols in terms of requirements. • assign criteria and measures for performance characteristics (e.g., bandwidth, latency) to protocols and determine them with system-specific programs. • decide which network technologies to use at different layers for a given application problem.
Contents	LV1: Operating systems: • Basic concepts computer architecture and operating systems • Processes and threads • Resource management (CPU(s), memory, file system, external HW) • User interfaces, including an introduction to the use of command lines (shells) Computer networks: • Computer network structure • Network components • Performance criteria (such as bandwidth, latency, fault tolerance) • Ethernet, Wireless LAN according to IEEE 802.11 • important network protocols (e.g. IP, TCP, UDP, http, https, DNS, DHCP) • Communication security: firewalls, virtual private networks LV2: • Practice for the contents of LV1
Study / Examination method	See document „Study and examination regulations (SPO)“ Possible bonus points: none
Types of media employed	Blackboard, beamer, demonstration, working on the computer
Reading list	• Christian Baun: Operating Systems / Betriebssyteme , Bilingual Edition, Springer Vieweg, 2020 • K.C.Wang: Systems Programming in Unix/Linux, Springer Nature, 2018 • Andrew S. Tanenbaum: Structured Computer Organization, 6th edition, Pearson

	• Andrew S. Tanenbaum: Modern Operating Systems, Pearson Education Limited, 5th edition • Christian Baun: Computernetze kompakt, Springer Vieweg, • Andrew S. Tanenbaum, Devid J. Wetherall: Computer Networks, 4th edition, Pearson (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_15, Data Science: Foundations of AI

Module Title	Data Science: Foundations of AI
Module Code	SDI_15
Course(s)	LV1: SDI_15a Data Science: Foundations of AI (seminar course) LV2: SDI_15b Practice for Data Science: Foundations of AI (practice / project)
Lecturer(s)	Prof. Dr. Klaus Radke
Module coordinator(s)	Prof. Dr. Klaus Radke
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 3, Winter semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 35h follow-up, 25h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS exercise/practices)
ECTS credits	5
Participation prerequisites	
Applicability of the module	The module can be used in the SD, SDI and MEDS degree programmes.
Module objectives / Learning Outcomes	**Knowledge**: The students ... • distinguish between supervised and unsupervised learning methods. • are familiar with machine learning methods for classification, regression and clustering problems. • can describe the data science process and assign the methods learnt to the respective phases. **Skills**: The students ... • apply data transformation methods, descriptive statistics and data visualisation. • carry out the training of a learning model. • evaluate the performance and informative value of a machine-trained model. **Competences**: Students are able to ... • work constructively in data science projects. • carry out small data science projects independently.
Contents	• Basic concepts of data science: data, models, features, labels, ... • Data preparation: introduction to python, data wrangling, feature engineering. • Description and visualization of data • Outlook on further data pre-processing methods • Supervised learning: linear regression, k-nearest neighbours classification, decision trees, random forests, neural networks • Unsupervised learning: k-means clustering • Evaluation metrics: accuracy, precision, recall, F1-score, ... • Methods for reproducibility of data experiments.
Study / Examination method	See document „Study and examination regulations (SPO)“ Possible bonus points: none
Types of media employed	blackboard, beamer, demonstration
Reading list	• James, Witten, Hastie, Tibshirani: An Introduction to Statistical Learning, Springer • Raschka et al.: Machine Learning with PyTorch and Scikit-Learn, Packt (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_16, Agile Development Methodologies

Module Title	Agile Development Methodologies
Module Code	SDI_16
Course(s)	LV1: SDI_16a Agile Development Methodologies (seminar course) LV2: SDI_16b Project for Agile Development Methodologies (practice / project)
Lecturer(s)	Prof. Dr. Timea Illes-Seifert
Module coordinator(s)	Prof. Dr. Timea Illes-Seifert
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 4, Summer semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 35h follow-up, 25h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar-based teaching, LV2: 2 SWS practice/internship)
ECTS credits	6
Participation prerequisites	Recommended: SDI_10 Foundations of Software Engineering, SDI_06 Project Management
Applicability of the module	The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • describe agile and non-agile software development processes. • describe the steps for agile release planning of a SW project. • apply planning steps of an agile SW development process • recall roles and responsibilities in an agile process and apply them • recall and
Contents	• Overview software development processes with their advantages and disadvantages • Characteristics of agility • SW project planning using agile development methods such as Scrum and Kanban as well as scaling options • Roles and responsibilities in agile projects • Communication and interaction in agile project teams • Agile release and sprint planning, Agile estimation and prioritization, acceptance criteria, user stories, tasks. • Tools for agile project planning and control
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points: none
Types of media employed	blackboard, beamer, demonstration
Reading list	• Mike Cohn: Succeeding with Agile: Software Development Using Scrum, Addison Wesley Signature Series • Esther Derby, Diana Larsen: Agile Retrospectives: Making Good Teams Great, Pragmatic Bookshelf • Ken Schwaber an Jeff Sutherland: Scrum Guide (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_17, Human Computer Interaction

Module Title	Human Computer Interaction
Module Code	SDI_17
Course(s)	LV1: SDI_17a Human Computer Interaction (seminar course) LV2: SDI_17b Practice for Human Computer Interaction (practice / project)
Lecturer(s)	Nicolai Ries
Module coordinator(s)	Prof. Dr.-Ing. Alexander Biedermann
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 4, Summer semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 35h follow-up, 25h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS exercise/practices)
ECTS credits	5
Participation prerequisites	
Applicability of the module	The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • name the basic patterns of human interaction with machines. • identify affordances and signifiers of interfaces. • describe the behaviour of interfaces via state models. • explain patterns and dark patterns in the interface design. • classify interfaces with respect to the use of patterns. • develop improvements of existing interfaces through patterns. • are the first to understand the importance of responsive design. • create exemplary interfaces according to established design principles. • know the importance of good interface design for the safety and ergonomics of operation.
Contents	• History and beginnings of human-machine interaction • Basic patterns of human interaction with machines • Sources of error and risk factors of the human-machine interface • Behavioural modelling of user interfaces via state machines • Pattern and anti-pattern in interface design • User interaction, user experience, hedonic qualities of human-machine interaction. • Design and design basics, skeuomorphism, flat design, organic design • Ergonomic design of human-machine interfaces, accessibility • Accessibility of human-machine interfaces in dependence of cultural imprint, previous experiences, age
Study / Examination method	See document „Study and examination regulations (SPO)“ Possible bonus points: none
Types of media employed	blackboard, beamer, demonstration
Reading list	• Lee, John D., Wickens, Christopher D., Liu, Yili, Boyle, Linda Ng: Designing for people an introduction to human factors engineering, CreateSpace • Don Norman: The Design of Everyday Things, Basic Books (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_18, IT Security

Module Title	IT Security
Module Code	SDI_18
Course(s)	LV1: SDI_18a IT Security (seminar course) LV2: SDI_18b Practice for IT Security (practice / project)
Lecturer(s)	Prof. Dr. Marie Oetzel
Module coordinator(s)	Prof. Dr. Marie Oetzel
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 4, Summer semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 30h follow-up, 30h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar-based teaching, LV2: 2 SWS practice/internship)
ECTS credits	5
Participation prerequisites	Recommended: SDI_14 Operating Systems and Networks
Applicability of the module	The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • understand the importance of IT security. • explain integrity, confidentiality, availability. • name cryptographic methods for encrypting information. • know measures for the protection of IT systems. • know the components of a Secure Software Development Lifecycle (SDLC). • perform threat analyses (threat modelling). • apply relevant security measures and tools within a CI/CD pipeline.
Contents	• Secure SDLC • Threat analysis (threat modelling) • Cryptographic techniques: Symmetric and asymmetric encryption, public key cryptography. • Attack vectors and techniques on IT systems. • Measures for the protection of IT systems.
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points for LV1: none Possible bonus points for LV2: student project
Types of media employed	Blackboard, beamer, demonstration, exercises
Reading list	• Howard and Lippner: The Security Development Lifecycle, Microsoft Press. • Long et al.: The CERT Oracle Secure Coding Standard for Java, Addison-Wesley. • Shostack: Threat Modeling: Designing for Security, Wiley. • Anderson: Security Engineering, Wiley. (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_19, Software Architecture and Design Patterns

Module Title	Software Architecture and Design Patterns
Module Code	SDI_19
Course(s)	LV1: SDI_19a: Software Architecture and Design Patterns (seminar course) LV2: SDI19b Practice for Software Architecture and Design Patterns (practice)
Lecturer(s)	Prof. Dr. Jóakim von Kistowski
Module coordinator(s)	Prof. Dr. Jóakim von Kistowski
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 4, Summer semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 35h follow-up, 25h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS practice)
ECTS credits	5
Participation prerequisites	Recommended: SDI_03 Foundations of Programming Technologies, SDI_04 Object-oriented Concepts, SDI_10 Foundations of Software Engineering, SDI_13 Collaboration, Quality and Test
Applicability of the module	The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students will ... • describe at least one modeling and design language (e.g., UML) and its content categories (such as diagram types). • describe architecturally significant non-functional requirements. • explain the process for creating and analyzing design patterns using a modeling and design language. • link elements to describe design patterns to represent algorithms. • implement design patterns in their own programs in a programming language. • list requirement criteria and rules for a software architecture (e.g., freedom from redundancy, uniqueness, cohesion). • distinguish software architecture patterns and architecture styles based on their ability to meet requirements. • analyze trade-offs between software architecture decision alternatives regarding technological and organizational impact. • classify software architectures in terms of meeting requirements. • form a software architecture using design patterns in a programming language.
Contents	LV1: • Conway's Law • Software quality scenarios • Software modeling • Modeling of software by means of e.g. UML • Object Design Patterns • Architectural decisions and their documentation • Software Architecture Patterns • Software Architecture Styles LV2: • Practice for the contents of LV 1
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points for LV1: none Possible bonus points for LV2: Completion of exercises with presentation
Types of media employed	Blackboard, beamer, demonstration, sample programs
Reading list	• M. Seidl, M. Scholz, Chr. Huemer, G. Kappel: UML @ Classroom: An Introduction to Object-Oriented Modeling, Springer, 2015 • Pascal Rocques: UML in Practice: The Art of Modeling Software Systems Demonstrated through Worked Examples and Solutions, 1st edition, Wiley, 2004 • Hassan Gomaa: Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures, Cambridge University Press, 2011 • Frank Buschmann, Kevlin Henney, Douglas C. Schmidt: Pattern-Oriented Software Architecture, Volume 4, A Pattern Language for Distributed Computing, Wiley (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_20, Parallel and Distributed Systems

Module Title	Parallel and Distributed Systems
Module Code	SDI_20
Course(s)	LV1: SDI_20a Parallel and Distributed Systems (seminar course) LV2: SDI_20b Practice for Parallel and Distributed Systems (practice)
Lecturer(s)	Prof. Dr. Martin Sträßer
Module coordinator(s)	Prof. Dr.-Ing. Alexander Biedermann
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 4, Summer semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 35h follow-up, 25h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS practice)
ECTS credits	5
Participation prerequisites	Recommended: SDI_14 Operating Systems and Networks, SDI_03 Foundations of Programming Technologies, SDI_04 Object-oriented Concepts, SDI_05 Data Structures and Algorithms
Applicability of the module	The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • describe requirements for Parallel and Distributed Systems (such as response time, data integrity, security, availability). • explain concepts to meet requirements for parallel and distributed systems and develop them to meet the requirements for a concrete system. • enumerate different programming interfaces in a language and evaluate them with respect to the set system requirements. • assign interfaces and elements of given operating and communication systems to the required concepts. • implement the assigned system components in a programming language. • explain parallel program processing. • divide concepts of parallel processing into categories. • describe concepts for process communication and synchronization. • configure operating and communication system components and services for distributed applications. • create components and interfaces for parallel processing. • interpret the influence of system parameters in distributed systems in their own program developments.
Contents	LV1: • Semaphores and mutexes • Sockets • Remote Procedure Calls • Inter-Process Communication • Threads • Processes • Process Management • Hardware parallelism • Multiprocessor systems LV2: • Practice for the contents of LV 1
Study / Examination method	See document „Study and examination regulations (SPO)“ Possible bonus points: none
Types of media employed	Blackboard, beamer, demonstration, sample programs
Reading list	• Comer, Douglas E.: Computer Networks and Internets: Global Edition, Pearson Education, • George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair: Distributed Systems: Concepts and Design, Addison Wesley (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_21, German as foreign Language A2

Module Title	German as a foreign language A2
Module Code	SDI_21
Course(s)	LV1: SDI_21a, German as a foreign language A2.1 LV2: SDI_21b, German as a foreign language A2.2 LV3: SDI_21c, Introduction to the German labour market
Lecturer(s)	Lyudmyla Englert, Prof. Dr. Benedict Kemmerer
Module coordinator(s)	N.N.
Teaching language(s)	German
Associated degree programmes, semester	Software Design International, Semester 1 (winter semester) and Semester 2 (summer semester)
Workload	Total effort: 270 h (of which: Contact hours: 120h, Independent study: 150h (of which: 50h preparation, 50h follow-up, 50h exam preparation))
SWS / teaching form	9 SWS (LV1: 4 SWS seminar teaching, LV2: 4 SWS seminar teaching, LV3: 1 SWS workshop and excursion)
ECTS credits	9 ECTS (4 ECTS + 4 ECTS + 1 ECTS)
Participation requirements	DAF A1
Applicability of the module	The module is open to non-native speakers in the Software Design International degree program.
Module objectives / Learning Outcomes	Competence of using the German language in an everyday-life and professional context; training of German skills in the four areas of listening, reading, speaking and writing according to the CEFR (Common European Framework of Reference for Languages) level of A0/A1-A2: • The student is able to identify and name relevant terms and vocabulary in level-adequate German (depending on the course of study). • The student is capable of level-adequate a) listening b) reading c) speaking d) writing • The student is enabled to prepare level-adequate language assignments in teams with other exchange students. • The student is able to engage in classroom discussions as well as simulations of typical everyday-life and business situations in level adequate German. • The student is capable of effectively interacting with other students in level-adequate German more confidently. • The student is prepared for German-taught lectures and for academic studies/international business dealings in his future academic/professional life.
Contents	Selected areas of everyday-life language and business/economic terminologies suitable for the level of the students (CEFR A0/A1-A2) The course SD_21c will introduce the students to the German labour market
Study / Examination method	See document „Study and examination regulations (SPO)“ Possible bonus points: none
Types of media employed	Blackboard, beamer, demonstration, sample programs
Reading list	• Schritte International, Hueber; the current edition will be communicated in the first session
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_22, German as Foreign Language B1

Module Title	German as a foreign language B1
Module Code	SDI_22
Course(s)	LV: SDI_22
Lecturer(s)	Lyudmila Englert
Module coordinator(s)	N.N.
Teaching language(s)	German
Associated degree programmes, semester	Software Design International, Semester 3, Winter semester
Workload	Total effort: 120 h (of which: Contact hours: 52h, Independent study: 68h (of which: 22h preparation, 24h follow-up, 22h exam preparation))
SWS / teaching form	4 SWS (LV1: 4 SWS seminar teaching)
ECTS credits	4 ECTS
Participation requirements	DAF A1
Applicability of the module	The module is open to non-native speakers in the Software Design International degree program.
Module objectives / Learning Outcomes	Competence of using the German language in an everyday-life and professional context; training of German skills in the four areas of listening, reading, speaking and writing according to the CEFR (Common European Framework of Reference for Languages) level of B1.1: • The students have enough language to get by, with sufficient vocabulary to express him/herself with some hesitation and circum-locutions on topics such as family, hobbies and interests, work, travel, and current events. • The student uses reasonably accurately a repertoire of frequently used "routines" and patterns associated with more predictable situations. • The student can keep going comprehensibly, even though pausing for grammatical and lexical planning and repair is very evident, especially in longer stretches of free production. • The student can initiate, maintain and close simple face-to-face conversation on topics that are familiar or of personal interest. Can repeat back part of what someone has said to confirm mutual understanding. • The student can link a series of shorter, discrete simple elements into a connected, linear sequence of points. • The student is prepared for German-taught lectures and for academic studies/international business dealings in his future academic/professional life.
Contents	Selected areas of everyday-life language and business/economic terminologies suitable for the level of the students (CEFR B1.1)
Study and Examination method	See document „Study and examination regulations (SPO)“ Possible bonus points: none
Types of media employed	Blackboard, beamer, demonstration, sample programs
Reading list	• Schritte International, Hueber; the current edition will be communicated in the first session • "Motive B1"- Kompaktkurs für Hochschulen im DaF-Bereich
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_23, Business Administration

Module Title	Business Administration
Module Code	SDI_23
Course(s)	Business Administration
Lecturer(s)	Prof. Dr. Gregor Weiche
Module coordinator(s)	Prof. Dr. Gregor Weiche
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 3, Winter semester
Workload	Total effort: 150 h (of which: Contact hours: 60h, Independent study: 90h (of which: 30h preparation, 30h follow-up, 30h exam preparation))
SWS / teaching form	4 SWS (seminar class/exercise/practices)
ECTS credits	5
Participation prerequisites	None
Applicability of the module	This module can be used in other engineering courses.
Module objectives / Learning Outcomes	The students ... • have basic knowledge of the field of general business administration, in particular the central economic processes in a company. • apply the basic knowledge from the field of business administration. • independently solve tasks and practical applications (e.g., case study) from a business perspective.
Contents	• Business goals (overview) • Business decisions (overview) • Operational functions, the value chain (overview) • Selected thematic specializations related to the study program
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points: none
Types of media employed	Blackboard, beamer, transparencies
Reading list	• Griffin, Ebert: Business Essentials, Pearson • Schulten: Integrated Business Processes, Espresso Tutorials (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_25, Mobile Applications and Development

Module Title	Mobile Applications and Development
Module Code	SDI_25
Course(s)	LV1: SDI_25a: Mobile Applications and Development LV2: SDI_25b: Practice for Mobile Applications and Development
Lecturer(s)	N.N.
Module coordinator(s)	Prof. Dr. Alison McNamara
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 6, Summer semester
Workload	Total effort: 150 h (of which: Contact hours: 60 h, independent study: 90 h (of which: 30 h preparation, 30 h follow-up, 30 h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS exercise)
ECTS credits	5
Participation prerequisites	Recommended: SDI_03 Foundations of Programming Technologies, SDI_04 Object-oriented Concepts, SDI_05 Data Structures and Algorithms, SDI_08 Multimedia Technology, SDI_17 Human Computer Interaction.
Applicability of the module	The module teaches basic knowledge and skills for the implementation of mobile applications. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • name and explain the similarities and differences of the central mobile target platforms iOS and Android and the associated app ecosystems • analyse and discuss the advantages and disadvantages of native tools and cross-platform frameworks. • name and distinguish the specific requirements and possibilities of mobile platforms and devices as a software platform and explain the associated life cycle of mobile applications • apply the design and technical specifications of the Android target platform in their own projects and work in teams to develop their own app prototypes and conceptualize example design solutions. • test, evaluate and optimize the jointly developed applications using the tools and procedures learned • present their own applications developed in the team and analyse the applications of other teams and developers in a structured and critical manner
Contents	LV1: • Features of iOS and Android as core mobile target platforms: Tools, Publishing Workflows, UI Guidelines • Specific requirements of mobile devices and platforms (performance, memory equipment, energy consumption, Internet connection ...) • Overview of cross-platform tools (React Native, Flutter, Xamarin ...) hybrid app frameworks and their advantages and disadvantages. • Data exchange of mobile applications with web databases via Restful APIs, local/persistent storage of data on mobile devices (SQLite) • Android SDK, Android Studio and the UI guidelines and design paradigms for Android (Material Design). • Security and rights management of apps, signing of apps, certificates and distribution through app stores LV2: • Basics and examples of designing and developing Android apps in Java using Android Studio. • Implementation of activity layout design and planning of activity lifecycles • Handling of intents, broadcasts, services and notifications • Targeted use of simulators and end devices in the mobile development process • Mobile application testing and debugging • Critical analysis and evaluation of the app implementations developed and presented during the exercises.
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points for LV1: none Possible bonus points for LV2: Processing of exercise projects with presentation
Types of media employed	Blackboard, beamer, worksheets, digital teaching content
Reading list	• Android Developer's Guide: developer.android.com • Material Design Guidelines: material.io (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_26, Web Technologies

Module Title	Web Technologies
Module Code	SDI_26
Course(s)	LV1: SDI_26a: Web Technologies LV2: SDI_26b: Practice for Web Technologies
Lecturer(s)	Prof. Dr. Martin Sträßer
Module coordinator(s)	Prof. Dr. Alison McNamara
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 6, Summer semester
Workload	Total effort: 150 h (of which: Contact hours: 60 h, Independent study: 90h (of which: 30h preparation, 30h follow-up, 30h exam preparation))
SWS / teaching form	4 SWS (LV1: 2 SWS seminar teaching, LV2: 2 SWS practice)
ECTS credits	5
Participation prerequisites	Recommended: SDI_11 Databases, SDI_18 IT Security
Applicability of the module	The module teaches basic knowledge of Web Technologies and competencies for the implementation of user-centred Internet applications. The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • name and explain the technical basics and framework conditions for browser-based applications (web standards, protocols/data exchange, client-server principle) • differentiate CSS and JavaScript frameworks and backend languages from each other and select and justify the adequate toolset for web applications • draw text in HTML, style the HTML markup using CSS, and create (simple and advanced) interactions using JavaScript. • make targeted use of web technologies within the framework of their own projects and program prototypes in teams and conceptualize exemplary design solutions, taking into account the best practices of web development • test, evaluate and present their own design solutions in a team and constructively discuss and criticize the solutions of other developers
Contents	LV1: • Overview of the Internet protocol family including HTTP/HTTPS, IPv4 and IPv6 • Client-server principle as a central element of Internet communication • Introduction web standards and frontend stack: HTML, CSS and JavaScript • CSS and JavaScript frameworks (Bootstrap, Angular.js, React) and WebApps (PWAs) publication. • Relational and non-relational web databases (MySQL/MongoDB ...) • Restful APIs and JSON as central interface technologies on the web • Backend languages (choice from PHP, Perl, Python, Ruby ...); Node.js (JavaScript in the backend) • Web-based content management systems: Drupal/Typo3/WordPress LV2: • User-centred and agile planning, conception, design, implementation and evaluation of own Internet applications by means of a selected framework • Targeted application and deepening of the competencies acquired in the lecture and in other modules in the context of web projects.
Study / Examination method	See document „Study and examination regulations (SPO)“ Possible bonus points: completion of computer lab exercises and a presentation
Types of media employed	Blackboard, beamer, worksheets, digital teaching content
Reading list	• Online courses on HTML, CSS and JavaScript www.w3schools.com (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_27, Software Development Project

Module Title	Software Development Project
Module Code	SDI_27
Course(s)	LV1: SDI_27a Software Development Project LV2: SDI_27b Seminar on Software Development Project
Lecturer(s)	Different
Module coordinator(s)	Prof. Dr. Timea Illes-Seifert
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 6, Summer semester
Workload	Total effort: 300 h (of which: Contact hours: 30h, Project: 200h, 40h seminar preparation (research, presentation), 30h exam preparation))
SWS / teaching form	2 SWS (LV1: Software Development Project, LV2: Seminar on Software Development Project)
ECTS credits	10
Participation prerequisites	SDI_03 Foundations of Programming Technologies, SDI_04 Object-oriented Concepts, SDI_05 Data Structures and Algorithms, SDI_06 Project Management, SDI_10 Foundations of Software Engineering, SDI_16 Agile Development Methodologies.
Applicability of the module	The module deepens knowledge in special disciplines of software engineering in the form of a seminar. Furthermore, the module pursues the objective of applying acquired knowledge and skills in a software development project and carrying it out in a team.
Module objectives / Learning Outcomes	The students ... • apply knowledge of software development, software quality and project management in the context of a concrete problem. • decide on a process model and determine which roles are necessary. • Plan and control all tasks needed to be performed as part of the software development project. • perform requirements analysis and derive the software architecture and the detailed design. • implement and test the application. • evaluate different solution alternatives in the area of requirements analysis, design, implementation and testing as in a team. • conduct team retrospectives on a regular basis. • research and present a selected, in-depth topic from the field of software engineering.
Contents	LV1: • Students carry out a software development project in a team. They learn which technical and non-technical challenges have to be mastered. • The implementation of the task requires the independent familiarization with development tools (IDEs, build tools, version management, etc.) and, depending on the specific topic, the analysis of various libraries or frameworks. LV2: • Research and presentation of a selected topic from the field of software engineering.
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points for LV1: none Possible bonus points for LV2: none
Types of media employed	blackboard, beamer, demonstration, practical work
Reading list	
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_28, Subject-specific compulsory Elective Module

Module Title	Subject-specific compulsory Elective Module
Module Code	SDI_28
Course(s)	Subject-specific compulsory Elective Module (seminar course / practice / project)
Lecturer(s)	Different
Module coordinator(s)	Representative for study planning
Teaching language(s)	Depending on the selected compulsory elective module
Associated degree programmes, semester	Software Design International, Semester 7, Winter semester
Workload	Total effort: 120 h (of which: Contact hours: 48h, Independent study: 72h (of which: 24h preparation, 28h follow-up, 20h exam preparation))
SWS / teaching form	4 SWS (seminar class/exercise/practices)
ECTS credits	4
Participation prerequisites	Depending on the selected compulsory elective module
Applicability of the module	As a rule, elective courses can be used in several degree programs. Detailed descriptions can be found in the module handbook of the elective subjects.
Module objectives / Learning Outcomes	Students acquire subject-specific knowledge of the selected subject areas or further subject-specific language skills depending on the language chosen. They apply the knowledge in simple applications from the subject-specific areas or are able to apply further language skills. The students are able to work on technical tasks from the module-specific areas, taking into account interdisciplinary aspects.
Contents	The contents are given in the description of the compulsory elective modules.
Study / Examination method	Depending on the compulsory elective module chosen
	Possible bonus points: Depending on the compulsory elective module chosen.
Types of media employed	Blackboard, transparencies, beamer (others depending on the chosen compulsory elective module)
Reading list	Depending on the selected compulsory elective module
Risk assessment for pregnant or breastfeeding students	Participation is possible after consultation with the lecturer

Module: SDI_29, Practical Training Semester

Module Title	Practical Training Semester
Module Code	SDI_29
Course(s)	Practical Training Semester
Lecturer(s)	-
Module coordinator(s)	Prof. Dr.-Ing. Konrad Doll
Teaching language(s)	German / English (depending on the chosen company or institution)
Associated degree programmes, semester	Software Design International, Semester 5, Winter semester
Workload	Total effort: 720 h
SWS / teaching form	0 SWS, Practical Training Semester
ECTS credits	24
Participation prerequisites	70 ECTS
Applicability of the module	The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students will... • know the operational work environment as well as typical activities of a software designer/computer scientist. • acquire knowledge of technical, organizational and business contexts in companies. • are able to perform adequately in companies, to fit into (project) teams and to cooperate professionally with other employees. • are able to understand the requirements of the company and align their own activities to these requirements. • work effectively and responsibly in companies as part of a team and with partners at different levels. • apply methods and knowledge from their previous studies to practical problems.
Contents	• Practical Training Semester in the company. • Content depends on the company and the specific task.
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points: none
Types of media employed	Practical work
Reading list	Depending on the company and the specific task at hand
Risk assessment for pregnant or breastfeeding students	Participation in the company internship is possible after consultation with the lecturer (individual risk assessment at the respective workplace).

Module: SDI_30, Practical Course

Module Title	Practical Course
Module Code	SDI_30
Course(s)	Practical Course
Lecturer(s)	Prof. Dr. Fabian Fürst
Module coordinator(s)	Prof. Dr. Barbara Sprick
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 5, Winter semester
Workload	Total time required: 90 h (of which: Contact hours: 30h, Independent study: 60h (of which: 20h preparation, 40h follow-up))
SWS / teaching form	2 SWS (seminar teaching)
ECTS credits	3
Participation prerequisites	None
Applicability of the module	The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students ... • provide technical terms of the working environment for future software designers. • recognize and name stake-holders in professional practice areas. • name basic framework conditions, regulations and laws in the work environment. • reflect processes in the work environment. • Assign recommendations for action and measures to work processes. • Identify challenges in the professional practice environment. • design measures for a holistic working environment.
Contents	• Workplace • Workplace design and environment • Work processes • Health protection and prevention • Quality requirements and standards • Process descriptions
Study / Examination method	See document „Study and examination regulations (SPO)“ None
Types of media employed	Blackboard, Projector, Demonstration, Practical work, Presentation
Reading list	• Marvin R. Weisbord: Productive Workplaces: Dignity, Meaning, and Community in the 21st Century, • Pfeiffer; 3. edition (9 Feb. 2012) • G. Anderson: Mastering Collaboration: Make Working Together Less Painful and More Productive, O'Reilly Media, 2019 (All books in their latest edition)
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_31, Practical Seminar

Module Title	Practical Seminar
Module Code	SDI_31
Course(s)	Practical Seminar
Lecturer(s)	Prof. Dr. Barbara Sprick
Module coordinator(s)	Prof. Dr. Barbara Sprick
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 5, Winter semester
Workload	Total effort: 90 h (of which: Contact hours: 30h, Independent study: 60h (of which: 24h preparation, 24h follow-up, 12h exam preparation))
SWS / teaching form	2 SWS (seminar teaching)
ECTS credits	3
Participation prerequisites	Practical Training Semester
Applicability of the module	The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students will ... • reflect, evaluate and analyse strengths and weaknesses of their activities in business practice and derive improvements for their own behaviour.
Contents	• Tasks in the Practical Training Semester
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points: none
Types of media employed	blackboard, beamer, demonstration, practical work
Reading list	-
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_32, Seminar for Bachelor Thesis

Module Title	Seminar for Bachelor Thesis
Module Code	SDI_32
Course(s)	Seminar
Lecturer(s)	Different
Module coordinator(s)	Prof. Dr.-Ing. Konrad Doll
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 7, Winter semester
Workload	Total effort: 60 h
SWS / teaching form	2 SWS, meetings with the supervisor, presentation of results
ECTS credits	2
Participation prerequisites	SDI_33 Bachelor thesis
Applicability of the module	The content of the module is aligned with the Software Design International degree program. The module is used exclusively for the Software Design International degree program.
Module objectives / Learning Outcomes	The students will ... • present, explain and defend their findings to a group in an understandable and activating way. • develop a detailed action plan and evaluate it regularly. • critically reflect on their own work. • research background information related to the topic. • engage in professional discourse, argue for their own position and discuss software design / computer science topics with experts in a results-oriented manner.
Contents	• In-depth development of your own topic including an evaluation concept using the design thinking method • Presentation of results in the form of a paper/presentation followed by discussion
Study / Examination method	See document „Study and examination regulations (SPO)“
	Possible bonus points: none
Types of media employed	blackboard, beamer, demonstration
Reading list	
Risk assessment for pregnant or breastfeeding students	Participation is possible

Module: SDI_33, Bachelor Thesis

Module Title	Bachelor Thesis
Module Code	SDI_33
Course(s)	-
Lecturer(s)	Different
Module coordinator(s)	Prof. Dr.-Ing. Konrad Doll
Teaching language(s)	English
Associated degree programmes, semester	Software Design International, Semester 7, Winter semester
Workload	Total effort: 360 h
SWS / teaching form	0 SWS
ECTS credits	12
Participation prerequisites	Activation by study office
Applicability of the module	The module bundles the scientific, methodological and practical skills acquired in the Software Design International course.
Module objectives / Learning Outcomes	In the context of the Bachelor thesis, students should independently apply the technical, methodological, social and personal skills they have acquired during their studies to a practice-relevant problem from the Software Design program on a scientific basis. The students will ... • work independently on a problem from the field of software design (computer science) within a given period of time frame using scientific methods under guidance and applying relevant theoretical and methodological knowledge. • demonstrate self, time and project management skills required to solve problems. • document the basic principles, methods used and results according to scientific standards. • deepen their problem-solving skills. They successfully transfer theoretical and methodological knowledge of software design to practical applications. • critically analyse the processes and challenges in information technology. • illuminate problems and approaches to solutions from an interdisciplinary perspective. • think in a networked way and strive for practical and sustainable solutions.
Contents	The student writes a written paper, usually presenting the results of a theoretical or experimental investigation or a practical development task. The processing steps are determined individually depending on the respective topic. Possible steps may include: • Familiarization with the topic and the current state of research/technology • Developing/selecting methods and techniques for solving problems and developing a solution concept • Application/implementation of the method and realization of the own concept • Evaluation, validation and assessment of the results
Study / Examination method	Bachelor Thesis
	Possible bonus points: none
Types of media employed	-
Reading list	
Risk assessment for pregnant or breastfeeding students	Participation is possible

Abbreviations

BA	Bachelor thesis
LN	Performance Record
S	Seminar
SU	Seminar teaching
Min.	Minutes
Prakt.	practical
Ü	Practice
Pr	Project / Praktikum
mE / oE	Successful / without success
SWS	Teaching hours per week
TN	Participation certificate
Pr. LN	Practical Performance record
schrP	Examination in writing
MdlP, mündlP	Oral examination
SPO	Study and examination regulations
AWPF	General science compulsory elective module
FWPF	Subject-specific compulsory elective module