fiw.thws

Faculty of Computer Science and Business Information Systems

Technical University of Applied Sciences Würzburg-Schweinfurt

Module Handbook

Bachelor Software Engineering (B. Eng.)

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Contents

1. semester

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1510600
Algebra

Module profile

Exam number
1510600

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Andreas Keller

Lecturer(s)
Prof. Dr. Andreas Keller

Applicability	BSED
Semester according to SPO	1. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
none

Recommended prerequisites for the participation in the module
Mastery of school maths

Content
General principles:
- Body of real numbers
- Principle of complete induction
- Introduction to the field of complex numbers

Linear algebra:
- Vector spaces (linear independence, basis and dimension)
- Matrices (calculating with matrices, trace and determinant, rank of a matrix)
- Linear systems of equations
- Gaussian algorithm
- Linear mappings
- Eigenvalues and eigenvectors
- Diagonalisation

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Written exam (sP) according to § 23 APO

Examination - length/format
90 minutes

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Students know basic mathematical concepts and terms that are used in computer science and technology.
- Students understand the importance of mathematical methods and their role in the development of computer science applications.
- Students apply mathematical techniques to solve practical problems in computer science and technology.
- Students analyse mathematical problems in order to identify and develop suitable solution strategies.
- Students evaluate different solution approaches and their effectiveness in the context of specific mathematical challenges.
- Students create complex mathematical models that can be used in real-world applications in computer science and engineering.

Literature

- Gramlich, Günter: Lineare Algebra - Eine Einführung; Fachbuchverlag Leipzig im Carl Hanser Verlag 2021
- Hartmann, Peter: Mathematics for Computer Scientists; Vieweg + Teubner, Wiesbaden 2019
- Papula, Lothar: Mathematics for Engineers and Scientists 1 and 2; Vieweg + Teubner; Wiesbaden 2024
- Schubert, Matthias: Mathematics for computer scientists; Vieweg + Teubner, Wiesbaden 2012
- Strang, Gilbert: Linear Algebra; Springer-Verlag, Berlin/Heidelberg/ New York 2003

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1510400
Databases

Module profile

Exam number
1510400

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Michael Rott

Lecturer(s)
Michael Rott

Applicability	BSED
Semester according to SPO	1. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
none

Recommended prerequisites for the participation in the module
None

Content
The module teaches the basic concepts and techniques of database development. The relational data model and the relational algebra are introduced as theoretical foundations. One focus is on database modelling, in particular the creation of entity-relationship models (ER models) and their conversion into relational schemas, taking normal forms into account. Introduction to the SQL language, including data manipulation, data queries and the definition of schemas and transaction management. Database development and administration is practised in practical exercises and projects during the semester.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Oral exam

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Students can explain basic concepts of data persistence and the differences between persistent and non-persistent data.
- Students can define the central terms of relational databases, such as relation, primary key, foreign key and normalisation.
- Students understand relational algebra and can apply simple operations to it.
- Students can explain the connection between conceptual, logical and physical data modelling and justify their importance for database development.
- Students are able to create entity-relationship models (ERM) for given use cases and convert these into relational schemas.
- Students can formulate and execute SQL queries for data manipulation (DML) and schema definition (DDL).
- Students can analyse existing database schemas and evaluate them with regard to redundancy, consistency and normal forms.
- Students are able to analyse technical information requirements and derive suitable data structures and queries from them.

Literature

- Michael Kofler (2024). Database Systems - The Comprehensive Textbook (2nd edition). Bonn: Rheinwerk Verlag GmbH
- Kemper, A., & Eickler, A. (2015). Database Systems - An Introduction (10th edition). Munich: De Gruyter Oldenbourg Verlag
- Elmasri, R., & Navathe, S. B. (2015). Fundamentals of database systems (7th edition). Munich: Pearson Studium
- Garcia-Molina, H., Ullman, J. D., & Widom, J. (2013). Database Systems: The Complete Book (2nd ed.). Upper Saddle River, NJ: Pearson
- Saake, G., Sattler, K.-U., & Heuer, A. (2011). Databases - Concepts and Languages (3rd ed.). Munich: Pearson Studium

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1510500
Introduction to Computer Science

Module profile

Exam number
1510500

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun

Applicability	BSED
Semester according to SPO	1. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
none

Recommended prerequisites for the participation in the module
none

Content
The module teaches the basics of computer science for students outside the core computer science programme.

- Information, information content, information coding, representation of numbers and characters, coding of text, dates, colour information
- Binary arithmetic, Boolean algebra and logic gates
- Models and modelling as a fundamental principle in computer science, abstraction, reduction, decomposition, aggregation
- Description of data structures with the extended Backus-Naur form
- Modelling of dynamic systems and their description with finite automata and state diagrams
- Formal languages, regular grammars and the word problem
- Other automaton models: Moore and Mealy automata
- The concept of algorithm, computability, halting problem, functionality and programming of Turing machines
- Basic algorithms for searching and sorting
- History of hardware development
- Structure and basic operation of a computer and microprocessor, Von Neumann architecture, Moore's law
- Structure and functioning of the Internet and World Wide Web
- Introduction to the HTML and Markdown languages
- Structure of distributed systems, client-server, peer-to-peer, blockchain, Git
- History of artificial intelligence, machine learning methods, regression, how neural networks work
- Data protection and ethics in computer science

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Oral exam

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Students remember basic concepts of information processing and can measure the information content of messages.
- Students understand data coding and basic methods of modelling within computer science.
- Students apply methods for describing data structures.
- Students analyse simple dynamic systems and describe them using state diagrams.
- Students create regular grammars and solve the word problem with the help of finite automata.
- Students analyse simple problems and create solutions with the help of Turing machines.
- Students remember important points in the history of computer science.

Literature

- Gumm, Heinz-Peter; Sommer, Manfred: Einführung in die Informatik, 10th edition, Oldenbourg, 2012.
- Ernst, Hartmut; Schmidt, Jochen; Beneken, Gerd: Grundkurs Informatik: Grundlagen und Konzepte für die erfolgreiche IT-Praxis, 8th edition, Springer Verlag, 2023

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1510100
IT Project Management and Business Administration

Module profile

Exam number
1510100

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Eva Wedlich

Lecturer(s)
Prof. Dr. Eva Wedlich

Applicability	BSED
Semester according to SPO	1. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
None

Content
The following topics are covered:
Introduction to project and project management
- Project organisation
- Project planning process
- Project costing
- Project control and monitoring
- Project completion

Basic concepts of business administration:
- The company
- The factors of production in business management
- Business objectives
- Business management key figures

Constitutive decisions of a business:
- Choice of location:
- Legal forms:

Business management functions:
- Procurement and purchasing
- Marketing and sales

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Written exam (sP) according to § 23 APO

Examination - length/format
90 minutes

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Students will be able to reproduce basic terms and concepts of IT project management and business administration.
- Students apply methods for developing project plans and carrying out a site analysis using practical examples.
- Students will be able to analyse project-related and business management data.
- Students assess the efficiency of project control mechanisms and make well-founded decisions on project costing and monitoring.
- Students evaluate alternative locations and strategic business management decisions.
- Students independently develop solution strategies for complex IT project management scenarios and business challenges.

Literature

- Vahs, D.; Schäfer-Kunz, J.: Einführung in die Betriebswirtschaftslehre; 9th edition; Schäffer-Poeschel, Stuttgart, 2025
- Wöhe, G.: Einführung in die allgemeine Betriebswirtschaftslehre; 28th edition; Vahlen; Munich, 2023
- Hanke, D.: Die 10 wichtigsten Methoden im Projektmanagement: Der schnelle, verständliche und bewährte Einstieg ins klassischen Projektmanagement, 2022
- Timinger, H.: Modernes Projektmanagement: Mit traditionellem, agilem und hybridem Vorgehen zum Erfolg, Wiley, 2024

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1510300
Introduction to Software Engineering

Module profile

Exam number
1510300

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
English

Organisation

Responsible lecturer
Prof. Dr.-Ing. Tobias Fertig

Lecturer(s)
Prof. Dr. Isabel John,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr.-Ing. Anne Heß

Applicability	BSED
Semester according to SPO	1. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
none

Recommended prerequisites for the participation in the module
none

Content
Software engineering covers all phases of software development, from the initial idea to the tested and delivered system. This module introduces the fundamental principles and concepts of software engineering, focusing on systematic development processes, requirements engineering, and basic modelling techniques. It provides the foundation for advanced modules on Analysis and Design and Software Quality.

Fundamentals of Software Engineering
- Objectives and principles of software engineering
- Characteristics and challenges of software projects
- Overview of software development activities and processes (agile vs. traditional)

Introduction to software (requirements) modelling with UML
- Introduction to UML Diagrams: Class Diagrams, Use Case Diagrams, Activity Diagrams, Sequence Diagrams

Software Development Processes & Team Collaboration
- Roles in software projects (e.g., Product Owner, Developer, Tester)
- Fundamentals of project organisation and documentation
- Cost and benefit estimation in software projects
- Agile processes (e.g., Scrum)

Quality Aspects
- Introduction to software quality (transition to "Software Quality")

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Written exam, Portfolio

The concrete length/format of the examination will be determined in the study plan.

Language of examination
English

Condition for the award of credit points
None

Learning outcomes

- Students are able to explain the principles of software engineering and assess their relevance.
- Students compare and justify the application of different software development processes.
- Students collect, model, and specify systematically software requirements using UML.
- Students use UML diagrams to structure and analyze software systems.
- Students evaluate software projects in terms of feasibility, cost, and quality.
- Students understand fundamental quality assurance methods.

Literature

- Sommerville, Ian: Software Engineering, Pearson, 2018
- Oestereich, Bernd: Analysis and design with UML 2.5, Oldenbourg, 2013/2020
- Rupp, Chris: UML crystal clear, Hanser, 2012
- McLaughlin: Object-oriented analysis and design from head to toe, O'Reilly, 2017
- Kecher, Christoph; Hoffmann-Elbern, Ralf; Will, Torsten T.: UML 2.5: Das umfassende Handbuch, Rheinwerk Computing, 2021

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1510200
Programming 1

Module profile

Exam number
1510200

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Tristan Wimmer

Lecturer(s)
Prof. Dr. Tristan Wimmer

Applicability	BSED
Semester according to SPO	1. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
None

Content
This module aims to teach students the basics of programming using the Python programming language. It introduces the basic concepts of programming languages and programming paradigms and creates the basis for further modules in the Software Engineering degree programme.
The following topics are covered:

- Elementary data types, data structures and operators
- Control structures: loops and conditional statements
- Programming with functions
- Introduction to object-orientated programming
- Introduction to the concept of inheritance
- Introduction to exception handling

In addition to these topics, this module demonstrates the appropriate structuring options for code, as well as documentation options for a clean and readable programming style. Furthermore, students are shown how best to encounter and solve problems.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Written exam (sP) according to § 23 APO

Examination - length/format
90 minutes

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Students identify and name the basic data types, data structures and operators and apply them in the Python programming language.
- Students explain how control structures such as loops and conditional statements control the flow of programmes and how these are implemented in Python.
- Students write simple Python programmes that use functions and parameter transfers to solve specific tasks and apply the divide and conquer principle.
- Students understand how to apply object-oriented programming in order to improve the structure and maintainability of a programme through encapsulation.
- Students design and implement an object-oriented programme in Python for a specific requirement, using the basic principles of inheritance.
- Students apply exception handling for incorrect input and data type incompatibility.

Literature

Häberlein, Tobias. Programming with Python: An Introduction to Procedural, Object-Oriented and Functional Programming. 1st ed. 2024. Berlin, Heidelberg: Springer Berlin Heidelberg, 2024. https://doi.org/10.1007/978-3-662-68678-2.

2. semester

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1511100 Algorithms and Datastructures

Module profile

Exam number
1511100

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr.-Ing. Tobias Fertig

Lecturer(s)
Prof. Dr.-Ing. Tobias Fertig, Prof. Dr.-Ing. Sebastian Biedermann

Applicability	BSED
Semester according to SPO	2. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
None

Content
The course deals with various complex algorithms and data structures in computer science in theory and practical application. Any programming language can be used to implement the solutions. The following main topics are covered in theory and practice as examples:

- The concept of algorithms, data structures
- Stacks, queues, lists (with optimisations)
- Graphs & various algorithms on graphs
- Different trees with their respective advantages and disadvantages
- Hashmaps and exploratory strategies
- Monte Carlo and Las Vegas algorithms
- Evolutionary algorithms
- Encryption algorithms and data protection
- Decentralised software and blockchain data structures

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Written exam (sP) according to § 23 APO

Examination - length/format
90 minutes

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Students name and characterise basic data structures and algorithms, including graph- and tree-based methods.
- Students explain typical application scenarios for algorithms using practical examples.
- Students select suitable data structures and algorithms for given problems.
- Students analyse the performance and scalability of the algorithms used.
- Students implement algorithms in a programming language and test their functionality.
- Students evaluate different solution approaches in terms of efficiency and applicability.
- Students independently develop algorithmic solutions for specific applications.

Literature

- Saake, Gunter; Sattler, Kai-Uwe: Algorithmen und Datenstrukturen, eine Einführung mit Java; 6th revised edition; dpunkt-Verlag; Heidelberg, 2020
- Cormen, T., Leiseren, C., Riverest, R., Stein, C.: Algorithms - An Introduction; 3rd edition; Oldenburg Verlag, 2010
- Fertig, Tobias; Schütz, Andreas. Blockchain the Comprehensive Guide. Rheinwerk Computing, 2024

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1510900
Analysis and Design

Module profile

Exam number
1510900

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Isabel John

Lecturer(s)
Prof. Dr. Isabel John,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr.-Ing. Anne Heß

Applicability	BSED
Semester according to SPO	2. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
none

Recommended prerequisites for the participation in the module
- Introduction to Software Engineering
- Programming 1

Content
This module builds on the fundamentals of "Introduction to Software Engineering" and teaches methods and techniques for the systematic analysis and structured design of software systems. Students learn to analyse complex requirements, design software architectures and apply model-based development approaches. The following topics are covered:

Requirements analysis and specification
- Elicitation, documentation and validation of requirements
- Application of use cases, user stories and scenarios
- Techniques for structuring and prioritising requirements

Fundamentals of design
- SOLID principles and clean code design
- Design patterns (e.g. factory, singleton, observer)
- Modelling and design of software architectures with UML
- Data modelling and interface design
- Detailed UML modelling (e.g. component diagrams, sequence diagrams, deployment diagrams)
- Introduction to architecture patterns such as MVC and standard architectures (layers, client-server, microservices)
- Abstraction levels: From logical to physical design

Tools and documentation
- Use of CASE tools for modelling
- Documentation techniques for architectural decisions

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio, Written exam

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Students are familiar with various aspects of requirements analysis
- Students analyse requirements systematically, structure them and document them.
- Students design software architectures based on design patterns.
- Students design database and API interfaces for applications.
- Students document and present architecture decisions in a well-founded manner.
- Students know different architecture patterns and apply them in a targeted manner.
- Students master methods of object-orientated design in software design.

Literature

- Sommerville, Ian: Software Engineering, Pearson, 2018
- Oestereich, Bernd: Analysis and design with UML 2.5, Oldenbourg, 2013/2020
- Rupp, Chris: UML crystal clear, Hanser, 2012
- McLaughlin: Object-oriented analysis and design from head to toe, O'Reilly, 2017

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1511000
Networks

Module profile

Exam number
1511000

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
English

Organisation

Responsible lecturer
Prof. Dr. Rolf Schillinger

Lecturer(s)
Prof. Dr. Rolf Schillinger

Applicability	BSED
Semester according to SPO	2. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
Computer science basics

Content
In this module, students gain a foundational understanding of networking technologies, their relevance for software engineering, and key security aspects of interconnected systems.
The course follows a layered approach, focusing on the four layers of the Internet Protocol Suite (TCP/IP) as the foundation of modern networking. Key topics include:

- Foundations: Basic networking concepts, ISO/OSI reference model as the conceptual grounding for modern networking
- TCP/IP concepts: Link layer and its protocols like Ethernet (IEEE 802.3) and Wi-Fi (IEEE 802.11), auxiliary protocols like (R)ARP, NDP, Internet layer with IPv4, IPv6, ICMP, IPsec, and routing basics (IGP, BGP), Transport layer with TCP and UDP, short introduction to QUIC, Application protocols with HTTP, HTTPS, DNS, and DHCP
- Networking devices, for example, NICs, hubs, switches, bridges, and routers
- Tools like Wireshark, network simulation via GNS3
- Mobile networking technologies such as 4G, LTE, and 5G
- Security considerations across all networking layers like ARP spoofing, IPSec, TLS attacks, firewalls, IDS and IPS

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Written exam (sP) according to § 23 APO

Examination - length/format
90 minutes

The concrete length/format of the examination will be determined in the study plan.

Language of examination
English

Condition for the award of credit points
None

Learning outcomes

- Students will be able to describe and explain the principles of layered network architectures
- Students will be able to define and explain the key functions of each of TCP/IP's layers
- Students will be able to design a scalable best-practice internetworking architecture for a fictional global company, incorporating appropriate link layer protocols, network hardware, subnetting, and routing protocols
- Students will be able to construct and operate a medium-sized TCP/IP network in a virtual lab environment
- Students will be able to evaluate a given network's structure and assess its effectiveness in meeting specified functional or performance requirements

Literature

- Tanenbaum, A. S., Wetherall, D. J., & Feamster, N. (2021). Computer networks (6th Global ed.). Pearson Education.
- Kurose, J. F., & Ross, K. W. (2021). Computer networking: A top-down approach (8th ed.). Pearson.
- Stevens, W. R., & Fall, K. R. (2011). TCP/IP illustrated, Volume 1: The protocols (2nd ed.). Addison-Wesley Professional.
- Stallings, W. (2017). Network security essentials: Applications and standards (6th ed.). Pearson.
- Grigorik, I. (2013). High performance browser networking: What every web developer should know about networking and web performance. O'Reilly Media.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1510800
Programming 2

Module profile

Exam number
1510800

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun

Applicability	BSED
Semester according to SPO	2. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
Programming 1

Content
The "Programming 2" module builds on the knowledge of Python and object-orientated programming (OOP) acquired in the first semester and introduces the programming language Kotlin. Both the language concepts of Kotlin and modern software development principles are taught. The focus of the module is on

- Kotlin syntax and concepts: variables, control structures, functions and lambdas
- Object-orientated programming in Kotlin: classes, objects, inheritance, interfaces and data classes
- Functional programming approaches: Higher-order functions, immutability and collections APIs
- Kotlin-specific concepts: Null-safety, extension functions, smart casts and coroutines
- Test-driven development (TDD) with Kotlin and unit tests
- Error handling with exceptions and idiomatic approaches in Kotlin
- Introduction to modern software development with Kotlin: use of build tools (Gradle), dependency management, simple application of design patterns
- Project work: realisation of a practical application using Kotlin
- Practical exercises and projects accompany the theoretical content so that students can independently develop Kotlin programmes and improve their quality through tests and code reviews.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Written exam (sP) according to § 23 APO

Examination - length/format
90 minutes

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes
After successfully completing the module, students will be able to

- Understand and apply basic and advanced Kotlin language concepts.
- Use object-orientated principles in Kotlin and design their own class hierarchies.
- Apply functional programming paradigms in Kotlin to write clean and efficient code.
- Understand and effectively use Kotlin's null-safety and type systems.
- Perform test-driven development (TDD) and unit testing with Kotlin to develop robust software.
- Implement efficient error handling using idiomatic Kotlin techniques.
- Manage Kotlin projects with modern build tools (e.g. Gradle) and integrate dependencies in a meaningful way.
- Design, implement and document a practical software solution in Kotlin.
- Analyse existing Kotlin code and check it for quality, readability and efficiency.
- Reflect on the differences and similarities between Python and Kotlin and weigh them up for different use cases.

Literature
- Kofler, Michael. Kotlin: The Comprehensive Handbook. 1st edition. Bonn: Rheinwerk, 2021.
- Szwillus, Karl. Kotlin: Introduction and practice. 1st edition. Frechen: mitp, 2020.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1510700
Project 1

Module profile

Exam number
1510700

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
1

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 0 hrs

Self-study: 135 hrs

Total: 150 hrs

Teaching format
Project

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun,
Prof. Dr. Isabel John,
Michael Rott,
Prof. Dr. Rolf Schillinger,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr. Frank-Michael Schleif,
Prof. Dr.-Ing. Sebastian Biedermann,
Prof. Dr. Tristan Wimmer,
Prof. Dr.-Ing. Anne Heß

Applicability	BSED
Semester according to SPO	2. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
Programming 1, IT Project Management and Business Administration, Introduction to Software Engineering, Fundamentals of Computer Science

Content
The students develop a console application in a group of three to four people according to given requirements. The application should contain a non-trivial algorithm (e.g. search, sorting, graph or optimisation algorithms) and work with files. The software consists of several modules, but without a comprehensive software architecture. The central topics include

- Modularisation of the code
- Efficient implementation of an algorithm
- Utilisation of version control (e.g. Git)
- Basic code documentation (README, function comments)
- Simple tests to validate the algorithm
- Simple task management and distribution within the team

Students are continuously supervised by a lecturer while working on the project. Supervision takes the form of regular meetings in which the progress of the work is discussed, as well as accompanying interim presentations in which the development of the project is reflected upon and further developed.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Presentation, Documentation

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes
After successfully completing the module, students have the following competences:

- Students will recall the basic concepts of software development and version control.
- Students understand the principles of modular software development and the structuring of projects.
- Students apply Git and version control in a team context.
- Students analyse non-trivial algorithms in terms of runtime and efficiency.
- Students evaluate different implementation approaches for a given problem.
- Students create a small, modular console application according to given requirements.
- Students create a technical presentation to present their solution.

Literature
Will be announced in the course depending on the specific projects and technology used.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1511200
Stochastics

Module profile

Exam number
1511200

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Patrik Stilgenbauer

Lecturer(s)
Prof. Dr. Patrik Stilgenbauer

Applicability	BSED
Semester according to SPO	2. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
none

Recommended prerequisites for the participation in the module
Algebra: Basic knowledge of linear algebra (in particular linear systems of equations, matrix algebra, vector spaces, scalar product), propositional and set algebra, combinatorics.
Programming I: programming logic, design of simple algorithms.
School knowledge of analysis assumed (differential and integral calculus).

Content
- Descriptive statistics: basic concepts, frequency distributions, location parameters, scattering parameters, linear correlation and regression analysis.
- Probability theory: result set, events, Kolmogorov's concept of probability, conditional probability and independence, discrete and continuous random variables, expected value and variance, law of large numbers, binomial distribution, hypergeometric distribution, Poisson distribution, exponential distribution, normal distribution, sums of random variables, central limit theorem.
- Inferential statistics: point and interval estimates, significance tests.
- Application and visualisation of stochastic methods using programming examples in Python or R.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Written exam (sP) according to § 23 APO

Examination - length/format
90 minutes

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes
After completing the module, students will be able to

- confidently apply basic concepts and terms of probability and statistics,
- calculate and interpret statistical ratios and probabilities and use them to describe data,
- select typical probability distributions and apply them to suitable, practical problems,
- analyse stochastic and statistical problems using logical and structured thinking and work on them in a solution-oriented manner,
- use the methods as a basis for further applications in software development, data science and machine learning.

Literature

- Bamberg, G., Baur, F. and Krapp, M.: Statistik, De Gruyter Oldenbourg, 2022.
- Bourier, G.: Descriptive Statistics, Springer Gabler, 2025.
- Bourier, G.: Probability theory and inferential statistics, Springer Gabler, 2018.
- Dreiseitl, S.: Mathematics for Software Engineering, Springer Vieweg, 2018.
- Henze, N.: Stochastics for Beginners, Vieweg + Teubner, 2011.
- Kurt, N.: Stochastics for Computer Scientists, Springer Vieweg, 2020.
- Teschl, G. and Teschl, S.: Mathematics for Computer Scientists - Volume 2 (Analysis and Stochastics), Springer Vieweg, 2014.

3. semester

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1511600
Backend Systems

Module profile

Exam number
1511600

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
English

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun

Applicability	BSED
Semester according to SPO	3. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
Programming 1 and Programming 2

Content

- Introduction to distributed systems, client-server, and peer-to-peer systems.
- Software architectures for backend systems (3-tier, hexagonal, monolithic vs. micro-service, event-driven)
- Frameworks to implement backend systems (e.g. Spring)
- Advanced database techniques, scalability, replication, sharding, ORM-tools, query caching, CAP theorem
- Protocols for remote procedure call, for example, GraphQL and Google RPC.
- Basics of the HTTP protocol and application in the form of Web APIs.
- Comprehensive introduction to the REST architecture principle: resources, URLs, CRUD, hypermedia, caching, security.
- Configuration of Web servers (Apache), load balancer, and public caches (nginx)
- Testing of backend systems, performance testing using JMeter, monitoring and logging
- Security aspects of network protocol and backend systems

In the traditional degree programme, the lecturer provides or agrees with the topics of the practical examples for the examination. In the dual study programme, the lecturer consults with the company on a task, ensuring practical relevance and feedback from the company.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio

The concrete length/format of the examination will be determined in the study plan.

Language of examination
English

Condition for the award of credit points
None

Learning outcomes

- The students understand the fundamental concepts and differences of distributed systems, including their architecture and communication models.
- The students analyse various software architectures for backend systems and evaluate their suitability for different use cases.
- The students apply advanced database techniques such as replication and sharding to enhance data availability and performance.
- The students implement a backend system using a framework like Spring, following best practices for configuration, deployment, and security.
- The students compare different protocols for remote procedure calls, such as GraphQL and Google RPC, assessing their strengths and limitations.
- The students design RESTful APIs by applying the principles of the REST architecture, focusing on resources, URLs, CRUD operations, and security strategies.
- The students evaluate the security aspects of network protocols and backend systems, proposing improvements based on best practices.

Literature

- Coulouris, J. Dollimore, and T. Kindberg, Distributed Systems: Concepts and Design (4th Edition) (International Computer Science). Boston, MA, USA: Addison-Wesley Longman Publishing Co, Inc, 2005.
- N. Biswas, Practical GraphQL: Learning Full-Stack GraphQL Development with Projects. Berkeley, CA: Apress, 2023.
- J. Webber, S. Parastatidis, and I. Robinson, REST in practice: hypermedia and systems architecture, 1st ed. in Theory in practice. Beijing Cologne: O'Reilly, 2010.
- L. Richardson and M. Amundsen, RESTful Web APIs, First edition, Second release. Beijing Cambridge Farnham Cologne Sebastopol Tokyo: O'Reilly, 2015.
- I. Dominte, Web API Development for the Absolute Beginner: A Step-by-step Approach to Learning the Fundamentals of Web API Development with .NET 7. Berkeley, CA: Apress, 2023.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1511700
Data Science

Module profile

Exam number
1511700

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
English

Organisation

Responsible lecturer
Prof. Dr. Frank-Michael Schleif

Lecturer(s)
Prof. Dr. Frank-Michael Schleif

Applicability	BSED
Semester according to SPO	3. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
Databases, Programming 1, Introduction to Software Engineering, Programming 2

Content
This module provides a foundational introduction to data science within software engineering, focusing on data-driven development, data management,
and practical analysis techniques. Students gain both theoretical knowledge and hands-on experience across the full data lifecycle.

Core Topics

Data Science & Data Literacy:
- Role of data in modern software systems
- Data-driven decision-making in engineering
- Data ethics, privacy, and compliance (e.g., GDPR)
- Data quality and bias awareness

Semi-Structured Data (XML & JSON):
- XML/JSON as flexible data formats
- Schema definitions (DTD, XML Schema)
- Querying techniques (XPath, XSLT, JSONPath)
- Real-world applications in software systems

Data integration & warehousing:
- Multidimensional modelling (star/snowflake schemas)
- ETL processes and data pipelines
- Data integration from relational sources and APIs
- OLAP and basic exposure to NoSQL/Big Data tools

Graph Databases & Networked Data:
- Basics of graph theory in data modelling
- Graph data structures and query languages (e.g., Cypher)
- Applications: recommendation engines, social networks

Practical Data-Driven Solutions:
- Design and implementation of software with integrated data flows
- Performance considerations in data handling
- Security, privacy, and compliance in data processing

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Written exam, Portfolio

The concrete length/format of the examination will be determined in the study plan.

Language of examination
English

Condition for the award of credit points
None

Learning outcomes
Upon successful completion, students will be able to:
Knowledge and Understanding:

- Explain core principles of data management and their application in software systems
- Describe key data models (relational, semi-structured, graph) and related technologies (XML, JSON, XPath, XSLT, Cypher)
- Understand data integration, ETL, warehousing, and NoSQL concepts
- Demonstrate awareness of data ethics, privacy, and regulatory frameworks

Skills and Competences:

- Design and query structured, semi-structured, and graph-based data
- Build ETL workflows and apply OLAP techniques for analytical processing
- Apply graph-based analysis for networked data
- Evaluate data quality, security, and compliance in practical contexts

Literature

- Skiena, S.S.; The Data Science Design Manual, Springer, 2017
- Robinson, I; Graph Databases 2nd Ed; O'Reilly Media; 2015
- Friesen, Jeff; Java XML and JSON; 2019
- Brian Knight, Professional Microsoft SQL Server 2014 Integration Services (Wrox Programmer to Programmer), Wrox, 2014
- Trevor Hastie, The Elements of Statistical Learning, Springer, 2009
- Ralph Kimball, Margy Ross, Warren Thornthwaite, Joy Mundy, Bob Becker: The Data Warehouse Lifecycle Toolkit, 2nd Edition, Wiley 2008

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1511800
Professional Skills

Module profile

Exam number
1511800

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Isabel John

Lecturer(s)
Prof. Dr. Peter Braun,
Prof. Dr. Isabel John,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr. Frank-Michael Schleif,
Prof. Dr.-Ing. Anne Heß

Applicability	BSED
Semester according to SPO	3. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
none

Recommended prerequisites for the participation in the module
none

Content
This module provides students with theoretical and practical knowledge and skills that can be applied in a professional working environment in a wide variety of areas.
Students learn and try out a range of methods, techniques and tools that are categorised into different focus areas. These include

- Learning and working techniques
- Scientific work
- Target group-orientated professional communication and
- Working in (international) teams

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Students apply methods for effective planning and structuring of their work processes
- Students name the basic principles of scientific work in computer science
- Students carry out a literature search and organise the results
- Students describe in detail the basics of designing effective scientific communication in the form of texts, presentations, posters and videos
- Students create scientifically orientated texts, presentations, posters and videos based on scientific standards
- Students learn relevant objectives, skills and best practices for planning, conducting and following up on various survey techniques (such as interviews, surveys)
- Students apply methods for effective communication in teams
- Students apply methods of interviewing to determine requirements

Literature
Will be announced in the lecture

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1511300
Project 2

Module profile

Exam number
1511300

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
1

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 0 hrs

Self-study: 135 hrs

Total: 150 hrs

Teaching format
Project

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun,
Prof. Dr. Isabel John,
Michael Rott,
Prof. Dr. Rolf Schillinger,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr. Frank-Michael Schleif,
Prof. Dr.-Ing. Sebastian Biedermann,
Prof. Dr. Tristan Wimmer,
Prof. Dr.-Ing. Anne Heß

Applicability	BSED
Semester according to SPO	3. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
Programming 1, Project 1, Programming 2, Introduction to Software Engineering, Analysis and Design, Algorithms and Data Structures

Content
In groups of four to five people, students develop an application focussing on the backend and the database. They apply elements of agile project management. The students are given a topic and determine the requirements together with the supervisor. A graphical user interface is not necessary or should be kept very simple. The software must have a recognisable software architecture (e.g. layers, pipeline, hexagonal). Quality assurance is carried out through unit tests. The students work with:

- Database connection with schema design (SQL)
- Advanced software design methods (e.g. UML, architecture patterns)
- Version control with Git (branching, pull requests, code reviews)
- Sprint-based development processes (e.g. Scrum, Kanban)
- Documentation includes architecture diagrams, user stories and automated API documentation.
- Quality assurance measures through automated unit tests

Students are continuously supervised by a lecturer while working on the project. Supervision takes the form of regular meetings to discuss the progress of the project and interim presentations in which the development of the project is reflected upon and further developed.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Documentation, Presentation

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes
After successfully completing the module, students have the following competences:

- Students understand basic architectural principles such as the layer model and modularisation.
- Students apply the layer model and modularisation concepts in the development of a distributed application with a database connection.
- Students apply agile methods to plan and implement tasks in sprints.
- Students analyse code changes as part of pull requests and carry out code reviews in a team.
- Students create structured software documentation for their application.
- Students create a technical project presentation to present their results.

Literature
Will be announced in the course depending on the specific projects and technology used.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1511500
Software Quality

Module profile

Exam number
1511500

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr.-Ing. Anne Heß

Lecturer(s)
Prof. Dr. Isabel John,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr.-Ing. Anne Heß

Applicability	BSED
Semester according to SPO	3. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
Introduction to Software Engineering
Analysis and Design

Content
This module provides a comprehensive introduction to the topic of software quality. It covers technical basics as well as process, management and usability topics.

The content includes the following topics:
- Software quality & software quality management
- Analytical quality assurance (software testing)
- Test process, activities and artefacts
- Strategies for test case determination (black-box testing, white-box testing)
- Types of testing (functional testing, regression testing, performance testing)
- Test-driven development
- Testing tools
- Bug tracking and reporting
- Static testing (reviews & static analysis)
- Constructive quality assurance using the example of usability / user experience
- Human-centred design and usability testing

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Students understand the terminology of software quality using a quality model
- Students can apply quality models to systematically determine quality requirements
- Students understand relevant activities and artefacts in the testing process
- Students can develop test strategies, define test cases and determine test coverage in the context of analytical quality assurance
- Students can select and apply test methods for functional and non-functional requirements.
- Students can establish suitable test management and customise it to the requirements of projects
- Students understand the importance of usability and user experience
- Students understand the objectives and techniques for constructive quality assurance based on the human-centred design process
- Students can apply the techniques in human-centred design to ensure good usability / user experience

Literature

- Liggesmeyer, P. (2009). Software Quality. Springer-Verlag.
- Spillner, A., & Linz, T.: Basiswissen Softwaretest: Education and training for Certified Tester - Foundation Level according to ISTQB® standard. dpunkt.verlag, 7th edition. 2024
- Schneider, K. (2012). Abenteuer Softwarequalität: Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement (2nd ed.). dpunkt.verlag.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1511400
System-Oriented Programming

Module profile

Exam number
1511400

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
English

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun

Applicability	BSED
Semester according to SPO	3. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
Programming 1 and Programming 2

Content
- Definition and meaning of system-oriented programming
- Using the command line of an operating system
- Shell programming using the example of Bash
- Working on remote computers with ssh
- Data processing on the command line with sed, awk, sort, jq
- Using AI assistance systems (e.g. GitHub Copilot, ChatGPT)
- Editing text documents with vim
- Using the version control system Git
- Introduction to the C programming language (syntax, data types, pointers, memory management)
- Build systems make, cmake, bazel
- System programming under Linux (system calls, handling files, processes)
- Debugging, profiling and performance optimisation (gdb, strace, ltrace, gprof)
- Security aspects of system-related programming and protection mechanisms
- Structure of the Linux operating system
- Processes, process management, scheduling
- Inter-process communication, race conditions, deadlocks, semaphores, Petri nets and deadlock detection, philosopher problem, producer-consumer problem
- Memory management, memory abstraction, partitioning, fragmentation, free memory management, virtual memory, page exchange algorithms
- Input and output, direct memory access, interrupts, hard disks, file systems for hard disks
- Backup methods, automation, data integrity
- Network communication and implementation of network protocols
- Hypervisor technologies, Docker containers, resource management

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio

The concrete length/format of the examination will be determined in the study plan.

Language of examination
English

Condition for the award of credit points
None

Learning outcomes

- The students understand the definition and principles of system-oriented programming, including its role in software development.
- The students demonstrate proficiency in using the command line of an operating system, applying advanced tools like sed, awk, sort, and jq for data processing tasks.
- The students develop shell scripts in Bash to automate system tasks and streamline operations effectively.
- The students utilize AI assistance systems, such as GitHub Copilot and ChatGPT, to improve coding efficiency and solve programming challenges.
- The students manage text documents using the vim text editor, employing advanced editing and configuration techniques.
- The students implement version control practices with Git to support collaborative software development workflows.
- The students program in C, focusing on syntax, data types, pointers, memory management, and use debugging and profiling tools like gdb, strace, ltrace, and gprof to analyse code performance.
- The students evaluate security aspects of system-related programming and apply protection mechanisms to ensure code security.

Literature

- D. J. Barrett, Efficient Linux at the command line: boost your command-line skills, First edition. Sebastopol, CA: O'Reilly, 2022.
- A. S. Tanenbaum and H. Bos, Modern operating systems, 4th ed. Boston: Prentice Hall, 2015.
- K. Hitchcock, Linux System Administration for the 2020s: The Modern Sysadmin Leaving Behind the Culture of Build and Maintain. Berkeley, CA: Apress, 2022.
- M. Kalin, Modern C Up and Running: A Programmer's Guide to Finding Fluency and Bypassing the Quirks. Berkeley, CA: Apress, 2022.
- K. Hitchcock, The Enterprise Linux Administrator: Journey to a New Linux Career. Berkeley, CA: Apress, 2023.
- J. Varma, Pro Bash: Learn to Script and Program the GNU/Linux Shell. Berkeley, CA: Apress, 2023.
- S. M. Palakollu, Practical System Programming with C: Pragmatic Example Applications in Linux and Unix-Based Operating Systems. Berkeley, CA: Apress, 2021.

4. semester

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 9999999
General Compulsory Elective

Module profile

Exam number
9999999

Duration
1 semester

Frequency
Every semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction

Language of instruction
German/English

Organisation

Responsible lecturer
Prof. Dr. Jochen Seufert

Lecturer(s)
Beate Wassermann

Applicability	BSED
Semester according to SPO	4. semester
Type of module	AWPM

Required prerequisites for the participation in the module according to the SPO
As a rule, none; exceptions are determined and announced by the Faculty of Natural Sciences and Humanities.

Recommended prerequisites for the participation in the module
none

Content
Selection of two general science electives (AWPF) (2 x 2.5 ECTS) or one AWPF (1 x 5 ECTS) from the range of subjects offered by the Faculty of Applied Natural Sciences and Humanities (FANG).

Range of subjects offered by the FANG in the areas of
- languages
- cultural studies
- Natural sciences and technology
- Politics, law and economics
- Education, psychology and social sciences
- Soft skills
- Creativity and art.

Courses whose content is already part of or directly related to parts of other modules of the degree programme are excluded from the FANG catalogue. The corresponding courses are marked with a blocking note in the FANG subject catalogue. The contents of the individual AWPFs are published on the FANG faculty's own homepage.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Written exam (sP) according to § 23 APO

Examination - length/format
90 minutes

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German/English

Condition for the award of credit points
None

Learning outcomes
The subject-specific learning objectives depend on the AWPF selected.
The students

- also acquire knowledge and competences that are not subject-specific but may be important for the desired career goal, such as special knowledge of foreign languages, natural sciences or social sciences
- analyse a wide variety of issues
- categorise subject-specific knowledge in an interdisciplinary context
- transfer what they have learnt to their current training
- have expanded their key competences and, where applicable, foreign language skills, which supports their personal development, also in intercultural terms
- are aware of their personal, social and ethical responsibilities.

Literature
depending on the selected AWPFs

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1512200
IT Security

Module profile

Exam number
1512200

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr.-Ing. Sebastian Biedermann

Lecturer(s)
Prof. Dr.-Ing. Sebastian Biedermann

Applicability	BSED
Semester according to SPO	4. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
None

Content
The "IT Security" module teaches basic concepts and practical approaches to securing IT systems and software applications. Students familiarise themselves with key terms, threat scenarios and attack vectors and understand the security objectives of confidentiality, integrity and availability. One focus is on cryptography, including symmetric and asymmetric encryption methods, digital signatures, certificates as well as public key infrastructures (PKI) and alternatives such as transport layer security (TLS). In addition, vulnerabilities in web applications, such as SQL injections, and in classic applications, such as buffer overflows, are analysed and countermeasures developed. The module also covers authentication and authorisation concepts, including multi-factor authentication, role and rights concepts and single sign-on (SSO). Another important component is threat modelling and risk analysis in order to systematically identify vulnerabilities and prioritise security measures. Practical exercises and case studies deepen the theoretical content and enable application to real-life scenarios. The module aims to equip students with the skills to design secure IT systems and effectively integrate security strategies into software development processes.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Written exam (sP) according to § 23 APO

Examination - length/format
90 minutes

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes
The students
- explain basic IT security terms, threat scenarios and attack vectors,
- apply this knowledge to different application contexts,
- explain the security objectives of confidentiality, integrity and availability,
- assess their relevance for IT systems and software projects,
- describe the functionality of symmetric and asymmetric encryption methods as well as digital signatures and certificates,
- use cryptographic procedures in suitable scenarios in a targeted manner,
- analyse vulnerabilities in web applications (e.g. SQL injection) and traditional applications (e.g. buffer overflows),
- develop suitable countermeasures to eliminate these vulnerabilities and justify their selection,
- explain modern authentication and authorisation concepts such as multi-factor authentication, role and rights concepts and single sign-on (SSO),
- and put these concepts into practice in security-relevant IT scenarios.

Literature
- Wolfgang Ertel: Angewandte Kryptographie, Hanser, 2019.
- Dafydd Stuttard, Marcus Pinto: The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Wiley, 2011.
- Tobias Klein: Buffer Overflows and Format String Vulnerabilities: Functionalities, Exploits and Countermeasures, dpunkt.verlag, 2005.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1512300
Machine Learning

Module profile

Exam number
1512300

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
English

Organisation

Responsible lecturer
Prof. Dr. Frank-Michael Schleif

Lecturer(s)
Prof. Dr. Frank-Michael Schleif

Applicability	BSED
Semester according to SPO	4. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
Data Science, Math modules, programming courses

Recommended prerequisites for the participation in the module
Math skills are crucial and are only briefly repeated on demand and in an initial refresher.
The students should have a reasonable knowledge in programming with python

Content
Machine learning (ML) is a core technology in software engineering, driving data analysis, automation, and AI. This module builds on prior data science knowledge to introduce essential ML concepts, models, and algorithms.

Students learn traditional AI approaches, core learning paradigms (supervised and unsupervised), and ethical considerations. A hands-on component emphasises practical skills in Python using libraries such as NumPy, Pandas, and Scikit-learn. By the end, students can build and evaluate ML models.

Module contents:

- Introduction to AI and ML (math refresher, AI history, symbolic vs. sub-symbolic methods, expert systems, classical models)
- Core ML Concepts (learning paradigms, prediction vs. discovery, ethics)
- Learning Foundations (loss functions, model evaluation, bias-variance trade-off)
- ML Algorithms (linear models, regularisation, clustering, decision trees)
- Practical Skills (Python, libraries, Jupyter, model tuning)
- Ethical & Societal Aspects (bias, fairness, privacy, AI impact)

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio

The concrete length/format of the examination will be determined in the study plan.

Language of examination
English

Condition for the award of credit points
None

Learning outcomes
Upon successful completion of this module, students will be able to:
Knowledge and Understanding:

- Explain AI/ML history (symbolic & sub-symbolic)
- Distinguish classical AI from modern ML
- Describe ML paradigms and applications
- Understand learning formalism: loss, risk, complexity
- Explain strengths, weaknesses, and efficiency of ML algorithms

Skills and Competences:

- Apply core ML algorithms (regression, classification, clustering)
- Use Python + libraries (Scikit-learn, Pandas, NumPy) for modelling
- Evaluate and compare models with metrics and validation
- Interpret results (accuracy, efficiency, robustness)
- Present findings via visualisations, reports, notebooks
- Assess ethical issues and bias in ML/AI

Literature

- Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, 2006.
- Kevin P. Murphy: Machine Learning: A Probabilistic Perspective, MIT Press, 2012.
- Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning, Springer, 2001.
- Stuart Russell, Peter Norvig: Artificial Intelligence: A Modern Approach, Pearson, 2022.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville: Deep Learning, MIT Press, 2016.
- European Commission: Ethical AI and Machine Learning Guidelines, 2023.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1512000
Mobile Systems

Module profile

Exam number
1512000

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
English

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun

Applicability	BSED
Semester according to SPO	4. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
None

Content
This module introduces students to the principles and practical aspects of mobile application development. It focuses on modern cross-platform technologies and best practices for building efficient, scalable, and user-friendly mobile applications.

- Introduction to Mobile User Interface Development with Flutter
- Overview of Flutter and the Dart programming language
- Designing responsive layouts, handling user input, and implementing navigation
- State Management in Mobile Applications
- Understanding stateful and stateless widgets in Flutter
- Exploring state management techniques such as Provider and Riverpod

- Introduction to React Native: Understanding the fundamentals of React Native and its component-based architecture
- Building cross-platform mobile apps using JavaScript and React principles
- Utilising React Native components and navigation techniques
- Integrating APIs and Asynchronous Data Handling in Mobile Apps

- Connecting mobile applications with backend services using REST APIs and GraphQL
- Handling asynchronous operations with Dart's Future & Streams and JavaScript's Promises & Async/Await
- Managing data fetching, caching, and error handling
- Testing and Debugging Mobile Applications

In the traditional degree programme, the lecturer provides or agrees with the topics of the practical examples for the examination. In the dual study programme, the lecturer consults with the company on a task, ensuring practical relevance and feedback from the company.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio

The concrete length/format of the examination will be determined in the study plan.

Language of examination
English

Condition for the award of credit points
None

Learning outcomes
After completing this module, students will be able to:

- Explain the fundamental concepts of cross-platform mobile development using Flutter and React Native, including their architectures, strengths, and limitations
- Compare and evaluate the architectural patterns of Flutter and React Native to justify technology choices for specific development scenarios
- Design responsive and user-friendly mobile interfaces using Flutter's widget system and React Native's component model
- Implement mobile applications with effective state management by using Provider, Riverpod, or React Context API
- Integrate backend services into mobile applications by consuming REST APIs and GraphQL, and by managing asynchronous data, caching, and error handling
- Develop and execute unit, widget, and integration tests using tools like Flutter's testing suite, Jest, and React Testing Library to ensure application correctness and stability

Literature
- Thomas Bailey, Alessandro Biessek: Flutter for Beginners: Cross-platform mobile development from Hello, World! to app release with Flutter 3.10+ and Dart 3.x. Packt, 2023.
- Sakhniuk, Mikhail, and Adam Boduch. React and React Native: Build Cross-platform JavaScript and TypeScript Apps for the Web, Desktop, and Mobile. Fifth edition. Birmingham, UK: Packt, 2024.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1511900
Project 3

Module profile

Exam number
1511900

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
1

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 0 hrs

Self-study: 135 hrs

Total: 150 hrs

Teaching format
Project

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun,
Prof. Dr. Isabel John,
Michael Rott,
Prof. Dr. Rolf Schillinger,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr. Frank-Michael Schleif,
Prof. Dr.-Ing. Sebastian Biedermann,
Prof. Dr. Tristan Wimmer,
Prof. Dr.-Ing. Anne Heß

Applicability	BSED
Semester according to SPO	4. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
Project 1, Project 2, Programming 1, Programming 2, Introduction to Software Engineering, Analysis and Design, Software Quality, Databases

Content
Students work in larger teams (5-6 people) on a complex software project with real functional and non-functional requirements that are to be determined by a customer through interviews. The students develop a distributed software system that integrates modern software engineering techniques. The team organises its tasks through an agile project management methodology and distributes roles in the team accordingly. The application could include:

- Microservices or a distributed architecture
- Database connection (SQL or NoSQL)
- A graphical user interface, for example as a web or mobile application
- Security aspects (e.g. authentication, authorisations, encryption)
- Performance and scalability optimisation
- Automated unit and integration tests for quality assurance
- The documentation follows industry standards with complete architecture, API and deployment documentation.

Students are continuously supervised by a lecturer while working on the project. Supervision is provided in the form of regular meetings to discuss the progress of the work, as well as interim presentations to reflect on and further develop the project.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Documentation, Presentation

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Develop a distributed software system with modern technologies
- Integrate IT security aspects into the software architecture
- Design and implement complex software solutions in teams
- Take on different roles in an agile team
- Apply advanced methods of agile project management, especially reviews and retrospectives
- Create complete software documentation
- Present your project convincingly in a final public presentation

Literature
Will be announced in the course depending on the specific projects and technology used.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1512100
Web Systems

Module profile

Exam number
1512100

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Rolf Schillinger

Lecturer(s)
Prof. Dr. Rolf Schillinger

Applicability	BSED
Semester according to SPO	4. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
None

Content
Over the last few decades, web systems have replaced many classic GUI applications that were previously programmed using GUI frameworks such as MFC/.NET/WPF, Qt, Swing and similar. Many advantages such as great platform independence, centralised maintenance and simple distribution to any number of clients have always been offset by considerable disadvantages such as limited offline capability or a lack of system integration. In practice, however, the advantages now outweigh the disadvantages in the majority of use cases. In this module, students learn about and utilise the various approaches and technologies on which modern web systems are based. This includes the following topics in particular:

- Fundamentals: HTML DOM and rendering processes in the browser, classic HTML templating systems such as Blade in PHP or Django in Python
- Reactivity and frameworks, reactivity in web applications, in particular reactive JavaScript frameworks such as React, component-based development, single page web apps, progressive web apps
- Performance and optimisation of page rendering, critical rendering path, server-side rendering
- Authentication and authorisation: basics of JWT and OAuth2, introduction to FIDO2 (Passkeys)
- Advanced topics in reactive JavaScript frameworks, state management (e.g. via Redux), tooling (Vite and Parcel), front-end testing
- Future technologies and hybrid approaches, WebAssembly, Electron & web-based desktop apps

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Written exam (sP) according to § 23 APO

Examination - length/format
90 minutes

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes
After successfully completing this module, students will be able to

- name the essential components of a modern web system and outline their interaction
- compare different HTML templating systems and differentiate between their most important features in order to evaluate their suitability for different scenarios
- implement given HTML layouts in a structured way using templating systems
- design a reactive web application and implement it as an SPA in React
- analyse the resulting web application for its rendering performance and optimise it in a targeted manner
- explain the JWT and OAuth2 specifications in detail and decide on this basis which solution is suitable for given requirements
- practically integrate both JWT and OAuth2-based authentication mechanisms in a sample React application

Literature

- Banks, A., & Porcello, E. (2020). Learning React: Modern Patterns for Developing React Apps (2nd ed.). O'Reilly Media.
- Google Developers (https://web.dev)
- Grigorik, I. (2013). High Performance Browser Networking: What every web developer should know about networking and web performance. O'Reilly Media.
- Mozilla Developer Network (https://developer.mozilla.org/de/)
- Müller, P. (2022). Getting started with HTML and CSS: Programming and designing websites (2nd ed.). Rheinwerk Verlag.
- Springer, S. (2022). React: The comprehensive handbook for modern front-end development. With practical examples on Redux, TypeScript, SSR and PWA (2nd ed.). Rheinwerk Verlag.

5. semester

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1512500
Internship Semester

Module profile

Exam number
1512500

Duration
1 semester

Frequency
Every semester

Credit hours (SWS)
1

ECTS-Credits (CP)
30.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 0 hrs

Self-study: 885 hrs

Total: 900 hrs

Teaching format
Internship

Language of instruction
German/English

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun

Applicability	BSED
Semester according to SPO	5. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
More than 90 ECTS points, of which 55 ECTS from modules from the first year of study and the Professional Skills module have been passed.

Recommended prerequisites for the participation in the module
In addition to the above-mentioned requirements, in particular: Software Quality, Backend Systems, Mobile Systems, Web Systems

Content
As part of the practical module, students are given the opportunity to apply and deepen the technical and methodological skills they have acquired during their studies in a real company environment. The practical phase includes a larger IT project in which students work independently through as many phases of the software development process as possible, including

- Requirements analysis and system specification
- Software architecture and design
- Implementation and testing
- System integration and introduction
- Documentation and quality assurance
- The project should have a duration of at least 12 weeks and address practical problems in the field of software engineering.

In addition, it is recommended that students familiarise themselves with various departments and areas of the company before the project in order to develop a holistic understanding of operational processes and cooperation in interdisciplinary teams. The practical phase is supervised by the university. The contact person for the supervised practical phase is the internship coordinator.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Presentation, Documentation

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German/English

Condition for the award of credit points
None

Learning outcomes
Students should achieve the following learning objectives during the practical semester:

- Acquire practice-orientated knowledge of operational procedures and processes and understand their relevance for software development.
- Work independently and autonomously in IT projects, under supervision and with increasing initiative.
- Apply the theoretical and practical skills acquired during the degree programme in real projects and link them to practical experience.
- Analyse practical requirements (e.g. customer wishes, technical requirements) and understand their significance for the development and implementation of software solutions.
- Design and implement solutions for operational processes or IT projects and evaluate their effectiveness.
- Work effectively in a team, understand role allocations and actively shape collaborative development processes.
- Recognise how software projects are embedded in the organisational structures of a company and actively integrate them into operational processes.
- Get to know the professional field of a software engineer in a professional environment and gain experience for your own career planning.
- Identify the right contacts in the company when problems arise and communicate with them in a goal-oriented manner.
- Experience best practices for professional software development, understand the high quality standards in companies and internalise excellence as an objective.
- Experience the dynamics and motivation in professional development teams and recognise the importance of communication, responsibility and team culture for the success of a project.
- Reflect on the meaning and impact of their own work in the company and recognise their contribution to value creation and the overall success of the organisation.

Literature

6. semester

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1513000
Advanced Software Testing

Module profile

Exam number
1513000

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
English

Organisation

Responsible lecturer
Prof. Dr. Steffen Heinzl

Lecturer(s)
Prof. Dr. Steffen Heinzl,
Prof. Dr. Tristan Wimmer

Applicability	BSED
Semester according to SPO	6. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
Programming 1 and 2, Introduction to Software Engineering, Analysis and Design, Software Quality

Content
The basics of testing

Motivation and goals of software testing
- Types of testing: black-, white- & grey-box testing
- Functional & non-functional tests
- Test coverage, test paths & test pyramid
- Test automation

Introduction to automated tests and success factors
- Test frameworks depending on the languages used
- Types of testing: record replay, scripted & keyword-driven testing
- Test architecture & design patterns

SOLID principles for test architectures
- 4-layer test concept: modelling, definition, execution, adaptation
- Important design patterns for testing
- Automated UI & API testing

Introduction to Selenium: Driver, PageObject Pattern, Identifiers
- Mocking with Wiremock for API testing
- Behaviour Driven Development (BDD)

Introduction to Cucumber & Gherkin
- Feature files, step files & scenario outlines
- Exploratory testing & DevOps integration

Exploratory testing methods and techniques
- Continuous testing, e.g. with Jenkins & DevOps pipelines

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio, Written exam

The concrete length/format of the examination will be determined in the study plan.

Language of examination
English

Condition for the award of credit points
None

Learning outcomes

- Students understand the basic concepts and principles of software testing, including test types, test coverage and the test pyramid.
- Students analyse different test architectures and evaluate their suitability for different software projects.
- Students apply test automation strategies and implement automated tests with JUnit, Selenium and Wiremock.
- Students develop a structured test architecture taking SOLID principles and design patterns into account.
- Students create Behaviour Driven Development (BDD) tests with Cucumber & Gherkin and integrate them into the development process.
- Students integrate automated tests into a DevOps process and implement continuous testing with Jenkins.
- Students evaluate the quality of tests using metrics such as test coverage, robustness and maintainability.
- Students experiment with exploratory test methods and apply suitable techniques for error detection.

Literature

- Essentials of Software Testing by Ralf Bierig, Stephen Brown, Edgar Galván, Joe Timoney, 2021, Cambridge University Press

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1512900
Clean Code and Design Pattern

Module profile

Exam number
1512900

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr.-Ing. Tobias Fertig

Lecturer(s)
Prof. Dr. Steffen Heinzl,
Prof. Dr.-Ing. Tobias Fertig

Applicability	BSED
Semester according to SPO	6. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
Students should bring their own project from, for example, a programming project, software development project, internship or similar.

Content
This module teaches widely used design patterns and clean code techniques that are essential for code quality and maintainability. The following contents are covered, among others:

- Naming, Functions, Comments, Formatting
- Objects and Data Structures
- Law of Demeter
- DTOs
- Exceptions, Don't return null
- Code Tangling and Scattering, Logging, AOP
- Fluent Interfaces, Internal DSLs
- Coding Dojos
- Dependency injection
- Traditional design patterns
- Modern web and mobile design patterns
- Refactoring strategies

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio, Written exam

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Students can develop more understandable code
- Students can develop more maintainable code
- Students can develop less error-prone code
- Students describe WHAT is to be achieved and not HOW at appropriate points in the code
- Students understand crosscutting concerns
- Students carry out refactorings independently
- Students understand design patterns and apply them

Literature

- Robert C. Martin (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Pearson Education.
- Fowler, M., Beck, K. (2018). Refactoring: Improving the Design of Existing Code. Addison-Wesley Professional.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1512800
Cloud Computing

Module profile

Exam number
1512800

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Rolf Schillinger

Lecturer(s)
Prof. Dr. Rolf Schillinger

Applicability	BSED
Semester according to SPO	6. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
None

Content
Since the first beginnings of virtualisation in IBM's mainframe operating systems in the 1980s, virtualisation technologies have evolved into today's cloud computing complex, which has brought about some ground-breaking innovations in the operation of IT services. Many of the features demanded of modern data centres today, such as maximum flexibility, agility and scalability, were either impossible or very difficult to implement before the widespread availability of cloud technologies.
This course deals with the most important basics, functionalities and characteristics of cloud computing, in particular with the following sub-areas:

- Fundamentals and differentiation
  - Virtualisation as the basis of cloud computing
  - Cloud operator models and their characteristics
  - HyperScaler (AWS, Azure, Google Cloud Platform)
- Virtual machines
  - Overview of common virtualisation technologies (KVM / QEMU, Proxmox, Hyper-V)
  - VM introspection / security of VMs
- Deployment in cloud environments and DevOps
  - Cloud in the context of DevOps
  - Automation of the deployment (pipelines, CI/CD)
  - Infrastructure as code
  - Scaling options
- Containerisation
  - BSD Jails, LXC
  - Docker containers
  - Orchestration of containers (Kubernetes)
- Overview of the PaaS offerings of some GigaScalers, e.g. AWS Elastic Beanstalk

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes
Upon successful completion of the module, students will be able to

- Describe the differences between traditional IT infrastructures and cloud deployments
- describe and differentiate between the various cloud operator models
- select suitable cloud technologies for given applications and plan their use
- create and configure a virtual machine on a provided Proxmox instance and deploy a backend application (consisting of Apache web server and database) on it
- Deploy this application containerised with LXC
- package the application in Docker containers and deploy it manually
- orchestrate these containers in a scalable manner using Kubernetes
- apply load testing methods to estimate the behaviour of the deployment under different load profiles

Literature

- Humble, J., & Farley, D. (2010). Continuous delivery: Reliable software releases through build, test, and deployment automation. Addison-Wesley.
- Liebel, O. (2023). Scalable Container Infrastructures: The Handbook for Planning and Administration. Rheinwerk Verlag.
- Merkel, D. (2023). Docker: The comprehensive handbook (current edition). Rheinwerk Verlag. ISBN 978-3836294722
- Stender, D. (2020). Cloud infrastructures: The handbook for DevOps teams and administrators. Rheinwerk publishing house.
- Tanenbaum, A. S., & Bos, H. (2014). Modern operating systems (4th ed.). Pearson International.
- Turnbull, J. (2021). The Docker book: Containerisation is the new virtualization (latest ed.). Self-published.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1512700
Data Protection and Ethics

Module profile

Exam number
1512700

Duration
1 semester

Frequency
Every summer semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Markus Oermann

Lecturer(s)
Prof. Dr. Markus Oermann

Applicability	BSED
Semester according to SPO	6. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO

Recommended prerequisites for the participation in the module
Introduction to Software Engineering
Analysis and Design
Software Quality

Content
The module provides knowledge and an understanding of the basic positions of digital ethics and the basic structures of data protection law. Students are enabled to apply what they have learnt to practical cases of software projects and to evaluate fundamental ethical and data protection issues during software development.

In the section on ethics, essential conceptual foundations of moral philosophy are explained. On the basis of established schools of ethics, the normative justification and, using problematic practical cases, the actual need for ethically sound software development and a corresponding ethos of those responsible are worked out. The examination of models for the integration of ethical considerations into development and system design processes illustrates how ethical principles can be incorporated into software development in practice.

In practice, questions of compliance with applicable data protection law are also of particular relevance. After an overview of its basic structures, the focus is on the requirements for technical and organisational data protection as well as the enforcement and consequences of legal violations. Relevant aspects of information security law are also included here. Finally, an overview of the European legal framework for artificial intelligence is provided.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Written exam (sP) according to § 23 APO

Examination - length/format
90 minutes

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Students have an overview of ethical requirements for software development and can map these in work processes.
- Students will be able to apply basic positions of digital ethics in practical cases and answer basic ethical questions in software development.
- Students have gained knowledge of the basic structures of data protection law and can evaluate basic questions of data protection compliance in software projects.
- Students have acquired knowledge of the basic structures of the legal framework for artificial intelligence.
- Students are able to communicate and engage in dialogue with the relevant experts for ethical and legal compliance in their future working environment.

Literature

- Schweppenhäuser, Gerhard (2021): Basic concepts of ethics. Ditzingen, Reclam: Chap. 2.3 - 3.
- Spieker gen. Döhmann, Indra (2025): Data protection law. Munich, Nomos.
- Windholz, Natascha et al. (2024): Praxishandbuch KI-VO. 1 edition. Munich, Hanser.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1512600
Project 4

Module profile

Exam number
1512600

Duration
1 semester

Frequency
Every semester

Credit hours (SWS)
1

ECTS-Credits (CP)
10.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 0 hrs

Self-study: 285 hrs

Total: 300 hrs

Teaching format
Project

Language of instruction
German/English

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun,
Prof. Dr. Isabel John,
Michael Rott,
Prof. Dr. Rolf Schillinger,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr. Frank-Michael Schleif,
Prof. Dr.-Ing. Sebastian Biedermann,
Prof. Dr. Tristan Wimmer,
Prof. Dr.-Ing. Anne Heß

Applicability	BSED
Semester according to SPO	6. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
Project 1, Project 2, Project 3. 100 ECTS must have been acquired.

Recommended prerequisites for the participation in the module
None

Content
Students work in larger teams (5-6 people) on a scientifically motivated and/or practical software project with complex functional and non-functional requirements. These requirements are determined either through interviews with an external stakeholder (e.g. company or research partner) or through scientific questions from current research topics. The project includes both the technical implementation and a scientific evaluation of the decisions made. The application could contain the following elements:

- Modern software architecture, e.g. microservices or distributed systems with database connectivity (SQL or NoSQL) with optimised query performance.
- A graphical user interface (web or mobile) with usability tests
- Security aspects (e.g. authentication, encryption, security audits)
- Automated tests and code quality analyses to ensure long-term maintainability

Students are continuously supervised by a lecturer while working on the project. Supervision takes the form of regular meetings in which the progress of work is discussed, as well as accompanying interim presentations in which the development of the project is reflected upon and further developed. In the traditional degree programme, the lecturers specify the topics of the practical examples for the examination or agree them with the students. In the dual study programme, lecturers coordinate an assignment with the company so that practical relevance and feedback from the company are guaranteed.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
error

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German/English

Condition for the award of credit points
None

Learning outcomes
After completing the module, students will be able to

- Develop a complex software system using modern technologies
- Work on and methodically evaluate a scientific problem from the field of software engineering
- Integrate IT security aspects into the software architecture
- Work with external stakeholders from industry or research and identify their requirements
- Apply agile project management methods, including time estimation and time tracking
- Create complete scientific software documentation according to academic standards
- Present results at a conference or in a scientific colloquium

Literature
Will be announced in the course depending on the specific projects and technology used.

7. semester

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1513400
Bachelor Thesis / Bachelor Seminar

Module profile

Exam number
1513400

Duration
1 semester

Frequency
Every semester

Credit hours (SWS)
1

ECTS-Credits (CP)
15.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 0 hrs

Self-study: 435 hrs

Total: 450 hrs

Teaching format
Seminar

Language of instruction
German/English

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun,
Prof. Dr. Isabel John,
Prof. Dr. Eva Wedlich,
Prof. Dr. Rolf Schillinger,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr. Frank-Michael Schleif,
Prof. Dr.-Ing. Sebastian Biedermann,
Prof. Dr. Tristan Wimmer,
Prof. Dr.-Ing. Anne Heß

Applicability	BSED
Semester according to SPO	7. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
120 ECTS points from the first four semesters, practical module, project work

Recommended prerequisites for the participation in the module
Fundamentals of scientific work in the Professional Skills module

Content
The Bachelor's thesis module consists of the Bachelor's thesis (12 CP) and the Bachelor's seminar (3 CP). The Bachelor's thesis comprises own studies and research on the state of the art and science in the respective subject area. The thesis must abstract from boundary conditions that are not technically based in nature, but result from the specific circumstances of the company. If software engineering solutions are required as part of the assignment, this usually means that prototypes are implemented, but does not include ensuring product features (including accompanying manuals etc.). In the Bachelor's seminar, the methods of scientific work are further deepened and practised.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Thesis, Presentation

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German/English

Condition for the award of credit points
None

Learning outcomes
With the Bachelor's thesis / Bachelor's seminar, students prove that they are capable of independently solving a challenging problem in the field of software engineering (possibly interdisciplinary) and that they have mastered the methodological and scientific principles of the subject and can adequately present the result.

Literature
depending on the topic; scientific literature must be intensively analysed, used and cited according to the topic

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1513900
FWPM 1

Module profile

Exam number
1513900

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar

Language of instruction
German/English

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun

Applicability	BSED
Semester according to SPO	7. semester
Type of module	FWPM

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
None

Content
The elective courses enable students to specialise in specific areas of software engineering. The focus is on current technologies, methods and concepts, which are taught in a practical manner and applied in projects. The content of the elective courses varies depending on the topics offered, but may focus on the following areas:

- Modern software architectures: in-depth study of microservices, event-driven architecture, domain-driven design (DDD) or cloud-native development
- Advanced programming concepts: Functional programming, metaprogramming, compiler construction or domain-specific languages (DSLs)
- Software quality and test strategies: test automation, continuous integration/continuous deployment (CI/CD), static code analysis and performance optimisation
- Software security: secure software development, penetration testing, security concepts in distributed systems
- Mobile and web technologies: development of progressive web apps (PWA), native vs. hybrid app development, modern frameworks and UI/UX optimisation
- AI and machine learning in software engineering: basics of machine learning, use of AI for software testing or code generation
- Database and persistence technologies: NoSQL databases, distributed data storage, optimisation of database queries

In the elective courses, students deal intensively with current challenges and solutions in software development. Practical exercises, project work and discussions supplement the theoretical content and promote an application-orientated approach to the respective subject areas.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German/English

Condition for the award of credit points
None

Learning outcomes
After successfully completing the elective course, students will be able to:

- Understand, apply and critically reflect on specialised concepts and technologies in a selected area of software engineering in order to solve complex problems in a targeted manner.
- Design and implement modern software solutions independently, taking into account best practices, current development methods and quality standards.
- Analyse innovative approaches and current research results from the chosen subject area and evaluate their practical relevance for software development.

Literature
Depending on the module selected.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1513901
FWPM 2

Module profile

Exam number
1513901

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar

Language of instruction
German/English

Organisation

Responsible lecturer
Prof. Dr. Peter Braun

Lecturer(s)
Prof. Dr. Peter Braun

Applicability	BSED
Semester according to SPO	7. semester
Type of module	FWPM

Required prerequisites for the participation in the module according to the SPO
None

Recommended prerequisites for the participation in the module
None

Content
The elective courses enable students to specialise in specific areas of software engineering. The focus is on current technologies, methods and concepts, which are taught in a practical manner and applied in projects. The content of the elective courses varies depending on the topics offered, but may focus on the following areas:

- Modern software architectures: in-depth study of microservices, event-driven architecture, domain-driven design (DDD) or cloud-native development
- Advanced programming concepts: Functional programming, metaprogramming, compiler construction or domain-specific languages (DSLs)
- Software quality and test strategies: test automation, continuous integration/continuous deployment (CI/CD), static code analysis and performance optimisation
- Software security: secure software development, penetration testing, security concepts in distributed systems
- Mobile and web technologies: development of progressive web apps (PWA), native vs. hybrid app development, modern frameworks and UI/UX optimisation
- AI and machine learning in software engineering: basics of machine learning, use of AI for software testing or code generation
- Database and persistence technologies: NoSQL databases, distributed data storage, optimisation of database queries

In the elective courses, students deal intensively with current challenges and solutions in software development. Practical exercises, project work and discussions supplement the theoretical content and promote an application-orientated approach to the respective subject areas.

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German/English

Condition for the award of credit points
None

Learning outcomes
After successfully completing the elective course, students will be able to:

- Understand, apply and critically reflect on specialised concepts and technologies in a selected area of software engineering in order to solve complex problems in a targeted manner.
- Design and implement modern software solutions independently, taking into account best practices, current development methods and quality standards.
- Analyse innovative approaches and current research results from the chosen subject area and evaluate their practical relevance for software development.

Literature
Depending on the module selected.

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1513300
Green IT

Module profile

Exam number
1513300

Duration
1 semester

Frequency
Every winter semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar-style instruction, Exercise

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr. Frank-Michael Schleif

Lecturer(s)
Prof. Dr. Frank-Michael Schleif

Applicability	BSED
Semester according to SPO	7. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
Given the advanced stage in the curriculum, students are expected to bring prior knowledge in software engineering, cloud computing, and machine learning, allowing for a deeper examination of sustainability challenges and solutions in IT.
Programming experience (e.g., Python) and familiarity with AI/ML frameworks (e.g., TensorFlow, PyTorch) are recommended but not mandatory.

Recommended prerequisites for the participation in the module
- Software Engineering
- Machine Learning
- data science

Content
Green AI & IT for Sustainable Digital Systems:
This module addresses the environmental impact of digital technologies and explores sustainable approaches to IT and AI. Students learn core principles of Green IT-such as energy-efficient hardware, data centres, and software-and emerging Green AI methods aimed at reducing the carbon footprint of machine learning. Topics include sustainable software engineering, AI model efficiency, and energy-aware computing. The module also covers process optimisation for sustainability, regulatory frameworks (e.g., EU AI Act, GDPR), and ethical considerations. Theory is complemented by practical strategies and expert-led enrichment lectures.

Core Topics:
- Green IT foundations: energy-efficient computing, sustainable infrastructure
- Green software engineering: efficient coding, lifecycle sustainability
- Green AI: low-energy model design, federated learning, inference optimisation
- Sustainable process modelling and green BPM
- Legal, regulatory, and ethical frameworks for sustainable IT/AI
- Emerging trends in sustainable computing and AI for global impact

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio, Written exam

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes
Upon successful completion of this module, students will be able to:

1. understand and explain the core principles of Green IT and Green AI, including their environmental and economic impacts.
2. describe and evaluate sustainable IT infrastructures and energy-aware AI systems, including data centres, cloud solutions, and efficient model architectures
3. interpret and apply relevant legal regulations, standards, and policies related to sustainability in IT and AI (e.g., EU AI Act, ISO 14001)
4. assess the carbon footprint and resource consumption of IT systems and AI models using established tools; optimise their efficiency through software and AI techniques
5. model and analyze IT processes with a focus on energy/resource efficiency and sustainability goals
6. develop comprehensive strategies for implementing sustainable IT and AI practices in organisational settings, integrating technical, regulatory, and business considerations

Literature

- Hilty, L. M., & Aebischer, B., ICT Innovations for Sustainability. Springer.
- Strubell, E., Ganesh, A., & McCallum, A. (2019), Energy and Policy Considerations for Deep Learning in NLP, https://arxiv.org/abs/1906.02243
- Bolón-Canedo, V., et al. (2023), Green Machine Learning: Advances in Energy-Efficient AI Models, Proceedings of ESANN 2023.
- European Parliament, AI Act: Environmental Regulations for Artificial Intelligence in the EU

fiw.thws
Faculty of Computer Science and Business Information Systems
Technical University of Applied Sciences Würzburg-Schweinfurt

Module: 1513100
Transfer colloquium

Module profile

Exam number
1513100

Duration
4 semester

Frequency
Every semester

Credit hours (SWS)
4

ECTS-Credits (CP)
5.0

Workload
Guided study time:
Sync. participation: 1 hrs
Async. participation: 3 hrs

Self-study: 90 hrs

Total: 150 hrs

Teaching format
Seminar

Language of instruction
German

Organisation

Responsible lecturer
Prof. Dr.-Ing. Sebastian Biedermann

Lecturer(s)
Prof. Dr. Peter Braun,
Prof. Dr.-Ing. Sebastian Biedermann

Applicability	BSED
Semester according to SPO	7. semester
Type of module	Compulsory module

Required prerequisites for the participation in the module according to the SPO
The module is only aimed at students in the dual study programme variant.

Recommended prerequisites for the participation in the module
None

Content
The transfer colloquium serves the exchange of students in the dual study programme variant and replaces one of the FWPM. Students on the normal version of the degree programme do not take part in this event. The event takes place over the third to seventh semester, with the exception of the practical semester. Under the guidance of a lecturer, the dual study programme students discuss the following topics:

- Importance of the university-company exchange
- Reflection on practical and study experiences in a professional context
- Exchange on challenges and solutions in dual study programmes
- Best-practice examples from everyday business life
- Strategies for time management and problem solving
- Application of theoretical content in the company
- Fundamentals and implementation of IT security in practice
- Documentation and presentation of project work

Examination

Required prerequisites for the participation in the examination according to the SPO appendix
None

Examination - type
Other exam (soP) according to §§ 26, 27 APO

Examination - length/format
Portfolio

The concrete length/format of the examination will be determined in the study plan.

Language of examination
German

Condition for the award of credit points
None

Learning outcomes

- Students are able to systematically reflect on their practical experience and draw conclusions for their professional development.
- Students develop strategies for solving problems that arise in the coordination between study and practice.
- Students are able to apply theoretical knowledge in practice in the company and thus optimise the transfer of knowledge between university and practice.
- Students are able to present and effectively communicate their practical experience in a clear and structured manner.
- Students understand the importance of IT security in the corporate context and can implement appropriate measures in their daily work.
- Students are able to create and present project documentation professionally in order to make their work transparent and comprehensible.
- Students learn to use feedback constructively in order to continuously improve their working methods and the cooperation between the university and the company.
- Students develop an understanding of their professional development opportunities and can make informed decisions for their career planning.

Literature
Will be announced in the seminar.