Module Guide
Applied Computer Sciences

Faculty Computer Science

Examination regulations 15.03.2025

Date: Wed 14.05.2025 13:27

MAI-01 THEORETICAL COMPUTER SCIENCE

Module code	MAI-01
Module coordination	Prof. Dr. Peter Faber
Course number and name	01.1 Theoretical Computer Science - Semantics, Computability and Complexity Theory 01.2 Theoretical Computer Science - Formal Languages and Compiler Construction I
Lecturer	Prof. Dr. Peter Faber
Semester	1, 2
Duration of the module	2 semester
Module frequency	annually
Course type	required course
Level	Postgraduate
Semester periods per week (SWS)	6
ECTS	8
Workload	Time of attendance: 90 hours self-study: 150 hours Total: 240 hours
Type of Examination	written ex. 90 min.
Duration of Examination	90 min.
Weighting of the grade	8/90
Language of Instruction	English

Module Objective

Theoretical computer science provides the fundamental ideas and computational models underlying every data processing system.

Understanding this theory is therefore essential for understanding verifiability and thus the security, correctness, and stability of a system.

In this module, students will learn the theoretical foundations of computer science, such as machine models with applications in complexity theory and computability theory, formal languages and their hierarchies, etc. They will gain an understanding of complexity estimation and verification options, as well as the basic functions of data processing systems.

Applicability in this and other Programs

Can be used for other suitable degree programs

Entrance Requirements

According to requirements of the respective program

Learning Content

The module consists of:

- Formal Languages and Compiler Construction I: This course explores the theoretical background behind formal languages and compiler frontends
- Semantics, Computability, and Complexity Theory: Here, theoretical (machine) models are used to formally define semantics, analyze computability, and estimate complexity.

Both subjects are closely related and complementary.

Teaching Methods

Lectures in form of seminars and practical excercises

Remarks

-

Recommended Literature

-

01.1 THEORETICAL COMPUTER SCIENCE - SEMANTICS, COMPUTABILITY AND COMPLEXITY THEORY

Objectives

The goal of this course is for students to understand and to apply formal theories and methods in the field of semantics, computability and theory of complexity.

Technical Competence:

- Application of formal calculation of the semantics of recursive functions
- Application of different induction methods to prove properties of programs
- Application of operational and axiomatic semantics to prove properties of programs
- Application of different models of computability
- Knowledge of the calculation of the complexity of different classes of problems and application of resulting consequences for software programming

Methodical Competences

- Application of mathematical proof concepts

Personal Competences

- Internalization of logical, mathematical thinking

Learning Content

- Semantics
  - Definition Semantics, History
  - Semantics of Recursive Functions (Fixed Point Theory)
  - Inductive Proofs
  - Operative Semantics
  - Axiomatic Semantics
- Computability
  - Definition
  - Unpredictable Functions
  - Turing Machines and their programming
  - Turing-Computability
  - LOOP-, WHILE-, GOTO-Computability
- Complexity Theory
  - Definition
  - O-Notation
  - Levels of complexity

Entrance Requirements

- Programming in an advanced programming language (e.g. C, C++, Java, C#)
- Mathematics of natural numbers (Induction)
- Basics of propositional and predicate calculus

Type of Examination

part of module exam

Methods

Lectures in form of seminars and practical excercises

Remarks

-

Recommended Literature

- John Longley, Lessons in Formal Programming Language Semantics, University of Edinburgh, 2003
- F.L. Bauer, H. Wössner: Algorithmische Sprache und Programmentwicklung, Springer Verlag 1984 (also availble in English)
- Rudolf Berghammer: Semantik von Programmiersprachen, Logos Verlag, 2001
- Juraj Hromkovic: Theoretische Informatik, Springer Verlag
- Uwe Schöning: Theoretische Informatik - kurz gefasst. Spektrum, 2008
- Hopcroft, Motwani, Ullman: Introduction to Automata Theory, Languages, and Computation, 3rd ed., Pearson 2014 (also available in German)

01.2 THEORETICAL COMPUTER SCIENCE - FORMAL LANGUAGES AND COMPILER CONSTRUCTION I

Objectives

In the course "Formal Languages and Compiler Construction I," students learn about formal languages (Chomsky hierarchy, etc.). They master concepts and constructions from automata theory and can reconstruct and understand corresponding proofs of language inclusions, computability, etc. They design small grammars and implement syntax-controlled translation schemes. They can classify the power of automata and, if necessary, justify them. The application of these theoretical foundations is demonstrated using compilers for programming languages.

On a professional level, students know and understand the information technology foundations of computability and decidability and are able to reconstruct and perform proofs themselves. On a professional and personal level, they can use this analytical methodology to compare and verify techniques and make statements about the solvability of a problem. The students have internalized the structured and analytical approach required for this.

Learning Content

- Introduction and Translators I

- Translators II / Formal languages I
- Formal languages II / III
- Lexical Analysis I / II
- Syntax analysis I / II
- Syntax analysis III / IV
- Syntax directed translation I/II
- Wrap-up, possibly further topics (e.g., optimizing compilers)

Entrance Requirements

Programming skills (ideally C) preferable, UNIX skills favorable

Type of Examination

part of module exam

Methods

Lectures in form of seminars and practical exercises

Remarks

-

Recommended Literature

- Compilers Principles, Techniques, and Tools; Aho, Lam, Sethi, Ullmann; 2nd edition; Addison-Wesley; 2007
- Engineering a compiler; Cooper, Torczon; 2nd Edition, Morgan Kaufmann 2012
- Introduction to Automata Theory, Languages, and Comoputation; Hopcroft, Motwani, Ullman; Addison-Wesley; 2001
- if necessary, further literature as specified by the lecturer

MAI-02 PRACTICAL COMPUTER SCIENCE

Module code	MAI-02
Module coordination	Prof. Dr. Peter Faber
Course number and name	02.1 Practical Computer Science - Formal Languages and Compiler Construction II 02.2 Practical Computer Science - Advanced Software Engineering 02.3 Practical Computer Science - Programming Lab
Lecturer	Prof. Dr. Peter Faber
Semester	1, 2
Duration of the module	2 semester
Module frequency	annually
Course type	required course
Level	Postgraduate
Semester periods per week (SWS)	6
ECTS	8
Workload	Time of attendance: 90 hours self-study: 150 hours Total: 240 hours
Type of Examination	Portfolio (With planned room)
Duration of Examination	90 min.
Weighting of the grade	8/90
Language of Instruction	English

Module Objective

Practical computer science introduces students to the practical application of theoretically grounded techniques. They are familiar with development methods and tools, as well as development processes of a system, and can apply their knowledge in practice using these tools.

Applicability in this and other Programs

Can be applied in other fields of study.

Entrance Requirements

None

Learning Content

The module consists of:

- Advanced Software Engineering: Here, students learn specific techniques and approaches of software engineering
- Programming lab: Here, students apply their software engineering skills in a real, small project, typically in teamwork
- Formal Languages and Compiler Construction II: This course explores practical aspects such as the backend of a compiler with optimization techniques, among others

Teaching Methods

Lectures in form of seminars and practical excercises

Remarks

-

Recommended Literature

-

02.1 PRACTICAL COMPUTER SCIENCE - FORMAL LANGUAGES AND COMPILER CONSTRUCTION II

Objectives

In the subject "Formal Languages and Compiler Construction II," students will learn the application of formal languages (Chomsky hierarchy, automata theory, etc.) in technology. They will apply concepts and constructs from automata theory, particularly in compiler construction. They will also learn the fundamentals of the compiler backend and optimizing compilation. They will understand the basic techniques of code generation up to code optimization and will be able to apply these techniques prototypically. Compilers for programming languages will be presented and examined in practical application.

From a technical perspective, students will become familiar with fundamental compiler technologies and understand the (in)efficiencies of programming language constructs. They will also gain technical insight into optimization methods. Both personally and professionally, students will gain insight into programming working methods, programming structures, and programming methodologies.

Learning Content

- IR generation I
- IR generation II

- RTEs I
- RTEs II
- Code generation I
- Code generation II
- Wrap-up, possibly further topics (e.g., optimizing compilers)

Entrance Requirements

"Formal Languages and Compiler Construction I"; Programming skills (ideally C) preferable, UNIX skills favorable

Type of Examination

part of module exam

Methods

Lectures in form of seminars with practical components.

Remarks

-

Recommended Literature

- Compilers Principles, Techniques, and Tools; Aho, Lam, Sethi, Ullmann; 2nd edition; Addison-Wesley; 2007
- Engineering a compiler; Cooper, Torczon; 2nd Edition, Morgan Kaufmann 2012
- if necessary, further literature as specified by the lecturer

02.2 PRACTICAL COMPUTER SCIENCE - ADVANCED SOFTWARE ENGINEERING

Objectives

Deepening of selected, key topics of software engineering.

Professional Competence:

- Students are familiar with and able to apply the agile Scrum method
- Students are familiar with selected UML diagrams in theory and practice
- Students are able to apply review methods to development results and projects

- Students are familiar with the specifics of testing object-oriented software

Learning Content

- Introduction
- Agile Methods in general
- Agile Methods - Scrum
- UML in general
- Selected UML diagrams in theory and practice
- Software Review Techniques, Intensive Reviews
- Special Features of testing object-oriented software

Entrance Requirements

- Knowledge of object-oriented programming
- Basic knowledge of software engineering

Type of Examination

part of module exam

Methods

Lectures in form of seminars and practical exercises, partly as group works, workshops

Remarks

-

Recommended Literature

- Peter Hruschka, Chris Rupp: Agile Softwareentwicklung mit der UML, Hanser Verlag, 2002
- Chris Rupp et. al: UML 2 Glasklar, Hanser Verlag, 2007
- Software Inspection, Tom Gilb and Dorothy Graham, Addison Wesley, 1993

02.3 PRACTICAL COMPUTER SCIENCE - PROGRAMMING LAB

Objectives

The students work (usually in teams) on a current programming task. They develop their own solutions to a given problem, as it might occur in a software company. The lecturer simulates the client and customer, or the interface to them.

The students gain a personal and professional understanding of a development process. They have used relevant tools and understood their function. On a personal and social level, they have experienced the challenges of a development process (ideally under changing requirements and problems, as is often the case today).

Learning Content

- Project presentation
- Needs-based: Introduction to computer science and programming techniques
- Project meetings
- Presentation of results

Entrance Requirements

- Lectures (Bachelor):
  - Foundations of Computer Science
  - Introduction to Programming
  - Software Engineering
- Simultaneous attendance of the module lecture "Theoretical Computer Science" as well as the other lectures of "Practical Computer Science"
- Programming and Software Development skills

Type of Examination

part of module exam

Methods

Students analyze a problem previously assigned by the lecturer, develop their own solutions, and implement them.
Feedback loops are agreed upon with the lecturer depending on the task. Support is provided through an e-learning system.

Remarks

-

Recommended Literature

According to the description of current programming frameworks on the web

MAI-03 SELECTED TOPICS OF EMBEDDED SOFTWARE DEVELOPMENT

Module code	MAI-03
Module coordination	Prof. Dr. Andreas Grzemba
Course number and name	03.1 Selected Topics of Embedded Software Development - Embedded Connectivity 03.2 Selected Topics of Embedded Software Development - Embedded Security
Lecturer	Stefanie Merz
Semester	1, 2
Duration of the module	2 semester
Module frequency	annually
Course type	required course
Level	Postgraduate
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	Portfolio
Weighting of the grade	5/90
Language of Instruction	English

Module Objective

The student acquires basic knowledge and skills in automotive ethernet communication and embedded security, structures, and reasoning. The student is competent to apply structured thinking and engineering thinking.

Technical Competences

Students know the absract concept of networking layers and their application in real-world hardware.

They are able to analyze and troubleshoot network communication issues in automotive/embedded environments

They apply critical thinking in evaluating different automotive network technologies

They are able to define virtual networks and interact with physical networks.

Methodical Competences
Students learn to systematically map and structure automotive network protocols

according to the OSI model, enabling them to categorize and analyze communication technologies efficiently. They apply a structured approach to implementing and testing network configurations in automotive applications.

Students are able to decide on the application of different analyzers in an embedded (netowrking) environment.

Personal Competences

Students obtain the ability to analyze and troubleshoot network communication issues in automotive environments. They will enhance their critical thinking skills by evaluating different automotive network technologies and their applications.

Applicability in this and other Programs

Module can be used in other degree programs, e.g. as a subject-specific elective module (FWP)

Entrance Requirements

Fundamentals of network technology and security

Learning Content

OSI-model

Automotive data comunication architecture

Standard IP protocolls

Automotive Ethernet physical layer

Data Link Layer: VLAN, TSN, AVB

Automotive application layer /SOME/IP

lab work at automotive multimedia geateway

Teaching Methods

Lectures in form of seminars and practical excercises

Remarks

-

Recommended Literature

Kirsten Matheus, Thomas Königseder; Automotive Ethernet; Cambridge University Press; 978-1-108-84195-5

03.1 SELECTED TOPICS OF EMBEDDED SOFTWARE DEVELOPMENT - EMBEDDED CONNECTIVITY

Type of Examination

part of module exam

03.2 SELECTED TOPICS OF EMBEDDED SOFTWARE DEVELOPMENT - EMBEDDED SECURITY

Type of Examination

part of module exam

MAI-04 SPECIAL MATHEMATICAL METHODS

Module code	MAI-04
Module coordination	Prof. Dr. Thorsten Matje
Course number and name	MAI-04 Special Mathematical Methods
Lecturer	Prof. Dr. Thorsten Matje
Semester	1, 2
Duration of the module	2 semester
Module frequency	annually
Course type	required course
Level	Postgraduate
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	written ex. 90 min.
Duration of Examination	90 min.
Weighting of the grade	5/90
Language of Instruction	English

Module Objective

The student acquires basic knowledge and skills in mathematical definitions, structures, and reasoning. The student is competent to apply structured thinking and mathematical reasoning.

Technical Competences

- Understanding of set theory ans probability
- Expertise in combinatorial methods, probability distributions, and statistical tests
- Application of different mathematical algorithms and methods
- Utilize data fitting techniques

Methodical Competences

- Application of mathematical concepts, such as proof concepts, and develop a structured approach to problem analysis
- Select appropriate probability distributions and estimation techniques for real-world problems

- Improve the capacity to formulate hypotheses, test assumptions, and interpret statistical data

Personal Competences

- Develop Logical reasoning and analytical thinking skills
- Enhance problem-solving abilities by working through challenging mathematical problems
- Cultivate precision and attention to detail when applying mathematical rules and formulas
- Internalization of logical, mathematical thinking

Applicability in this and other Programs

This module lays the basics in understanding contexts of higher mathematics.

Entrance Requirements

none

Learning Content

1. Set Theory and Probability
    - Random Experiments and Events
    - Set Theory
    - Probability
    - Laplace Experiment
    - Kolmogoroff's Axioms
    - Conditional Probability
    - Stochastic Independence
    - Addition Rule
    - Multiplication Rule
    - Probability Tree
    - Bayes' Theorem
    - Combinatorics
    - Variation

- Combination
- Permutation

- Probability Distributions
  - Random Variables
  - Bernoulli Experiments
  - Distribution of a Random Variable
  - Expected Value
  - Probability Distributions
  - Laws for Discrete Distributions
  - Laws for Continuous Distributions
  - Binomial Distribution
  - Normal Distribution
  - Standard Normal Distribution
  - z-transformation
  - Hypergeometric Distribution
- Statistical Tests
  - Sample Distribution of Characteristic Values
  - Estimation Procedures
  - Simple Point Estimation
  - Confidence Intervals
  - Degrees of Freedom
  - Confidence Interval for a Proportion
  - Central Limit Theorem
  - Method of Statistical Tests
  - ANOVA
  - Chi-Squared-Test
  - Error Analysis

- Important Distributions
  - Poisson Distribution
  - Negative Binomial Distribution
  - Geometric Distribution
  - Discrete Uniform Distribution
  - German Tank Problem
  - Uniform Distribution
  - Exponential Distribution
  - Pareto Distribution
  - Logistic Distribution
  - Weibull Distribution
- Monte Carlo Simulation
  - Business Planning Example
  - Markov Chains and Metropolis?Hastings Algorithm
- Fitting Data
  - Least Squares Method
  - Linear Least Squares
  - Nonlinear Least Squares

Teaching Methods

Lectures and exercises

Remarks

-

Recommended Literature

- Probability with Statistical Applications; Schinazi, Rinaldo B.; 2022; Springer International Publishing Birkhäuser
- Foundations of Modern Probability; Kallenberg, Olav; 2021; Springer International Publishing Springer

- One thousand exercises in probability; Grimmett, Geoffrey; 2020; Oxford University Press
- Introduction to Probability and Random Variables; Gazi, Orhan; 2023; Springer Nature Switzerland Springer
- Applied Probability; Lange, Kenneth; 2024; Springer US Springer
- Probability : An Introduction Through Theory and Exercises; Baldi, Paolo; 2023; Springer International Publishing Springer
- Probability Theory I : Random Variables and Distributions; Pascucci, Andrea; 2024; Springer Nature Switzerland Springer
- Practical statistics for the analytical scientist : A bench guide; Ellison, Stephen L R.; 2009; Royal Society of Chemistry
- Monte-Carlo simulation : an introduction for engineers and scientists; Stevens, Alan; 2023; CRC Press
- Regression and Fitting on Manifold-valued Data; Adouani, Ines; 2024; Springer Nature Switzerland Springer
- Applications of Linear and Nonlinear Models : Fixed Effects, Random Effects, and Total Least Squares; Grafarend, Erik W.; 2022; Springer

MAI-05-09 ELECTIVE COURSES 1 - 5

Module code	MAI-05-09
Module coordination	Prof. Dr. Peter Faber
Course number and name	MAI-05-09 Electives 1 - 5
Lecturer	Dozierende der ausgewählten Wahlpflichtfächer Lecturer of the chosen Electives
Semester	1, 2
Duration of the module	2 semester
Module frequency	annually
Course type	elective course
Level	Postgraduate
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	Examination form of the chosen module
Weighting of the grade	5/90
Language of Instruction	Course dependent

Module Objective

In this dynamic module, students are empowered to tailor their educational journey by selecting an elective from a diverse array of existing university courses or engaging in student research projects facilitated by esteemed university lecturers. This customizable approach ensures that students have the flexibility to align their academic pursuits with their unique interests, career aspirations, and individual learning needs.

There are three main goals in this module to the benefit of each student:

- The first goal is to fill knowledge gaps of the student (individuality).
- The second goal is to acquire knowledge in current and different upcoming topics of computer science (flexibility).
- As a third goal, students should be able to advance in individual higher-level topics (specialization).

Possible subjects are regularly presented, discussed and selected in a corresponding e-learning course that defines necessary prerequisities for certain courses and allows for additional personal discussions.

Methodological skills

A key objective of this module is to provide students with the opportunity to explore current and emerging topics within the field of computer science. By selecting electives that delve into diverse areas of research and innovation, students gain exposure to cutting-edge advancements, methodologies, and technologies shaping the future of computational science.

The exact methodological skills depend greatly on the selected subject.

Social skills

Most of the proposed subjects will have the students prepare presentations and / or work in groups on projects, where they will work together with each other for a successful participation.

The exact social skills supported by the module, however, depend greatly on the selected subject.

Personal skills

Students identify specific knowledge gaps within their academic foundation. The primary objective is to customize the elective selection process to address these individualized learning needs effectively. Reflecting on their individual situations and needs, they make a well-informed decision on their further studies.

Professional Skills

Within this module, students have the opportunity to cultivate a range of professional skills essential for success in their academic and professional endeavors. They develop strong communication skills through interactions with the study coordinator, faculty members, and peers, articulating their academic goals, and discussing potential electives or research projects. Additionally, students enhance their organizational skills as they navigate the elective selection process. supported by the module, however, greatly depend on the selected subject.

Applicability in this and other Programs

not applicable

Entrance Requirements

according to chosen elective

MAI-05-09 ELECTIVES 1 - 5

Type of Examination

Examination form of the chosen module

MAI-10 ELECTIVE 6

Module code	MAI-10
Module coordination	Prof. Dr. Peter Faber
Course number and name	MAI-10 Elective 6
Lecturer	Dozierende der ausgewählten Wahlpflichtfächer Lecturer of the chosen Electives
Semester	3
Duration of the module	1 semester
Module frequency	each semester
Course type	compulsory course
Level	Postgraduate
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	Examination form of the chosen module
Weighting of the grade	5/90
Language of Instruction	English

Module Objective

In this dynamic module, students are empowered to tailor their educational journey by selecting an elective from a diverse array of existing university courses or engaging in student research projects facilitated by esteemed university lecturers. This customizable approach ensures that students have the flexibility to align their academic pursuits with their unique interests, career aspirations, and individual learning needs.

There are three main goals in this module to the benefit of each student:

- The first goal is to fill knowledge gaps of the student (individuality).
- The second goal is to acquire knowledge in current and different upcoming topics of computer science (flexibility).
- As a third goal, students should be able to advance in individual higher-level topics (specialization).

Possible subjects are regularly presented, discussed and selected in a corresponding e-learning course that defines necessary prerequisities for certain courses and allows for additional personal discussions.

Methodological skills

A key objective of this module is to provide students with the opportunity to explore current and emerging topics within the field of computer science. By selecting electives that delve into diverse areas of research and innovation, students gain exposure to cutting-edge advancements, methodologies, and technologies shaping the future of computational science.

The exact methodological skills depend greatly on the selected subject.

Social skills

Most of the proposed subjects will have the students prepare presentations and / or work in groups on projects, where they will work together with each other for a successful participation.

The exact social skills supported by the module, however, depend greatly on the selected subject.

Personal skills

Students identify specific knowledge gaps within their academic foundation. The primary objective is to customize the elective selection process to address these individualized learning needs effectively. Reflecting on their individual situations and needs, they make a well-informed decision on their further studies.

Professional Skills

Within this module, students have the opportunity to cultivate a range of professional skills essential for success in their academic and professional endeavors. They develop strong communication skills through interactions with the study coordinator, faculty members, and peers, articulating their academic goals, and discussing potential electives or research projects. Additionally, students enhance their organizational skills as they navigate the elective selection process. supported by the module, however, greatly depend on the selected subject.

Applicability in this and other Programs

not applicable

Entrance Requirements

according to chosen elective

Learning Content

During the Master Applied Computer Science program, the student must select 1 elective subject.

The Applied CS master program features a list of default electives.The student may choose should come from the following list of **default electives**:

Possible electives (SS)

- Datacenter Network Programming (AIX, Kassler, 4 SWS, 5 ECTS)
- Advanced Driver Assistance Systems (ASE-1, Limbrunner, 4 SWS, 5 ECTS)
- ChatGPT et al.: Generative AI with Transformers (AI-X, Fischer, 4SWS, 5 ECTS)
- Computational Logic (AIN-B, Ewender, 4SWS, 5ECTS, additional task required)
- Advanced Programming Techniques (ET-M, Wölfl, 4 SWS, 6 ECTS)
- Data Visualization (LSI-M, Torkler / Valdez, 4 SWS, 5 ECTS)
- Bioinformatics -- Algorithms and Data Structures (LSI-M, Torkler, 4 SWS, 5 ECTS)
- Artificial Intelligence and Software Development (AIDS-M, Ewender, 4 SWS, 5 ECTS)
- Programming in C++ (Faber, 6 ECTS, 4 SWS, Virtuelle Hochschule Bayern [VHB])
- Machine Learning for Engineers I (Eskofier et al. [VHB], 4SWS, 5 ECTS)
- Tele-Experiments with Mobile Robots (Nüchter [VHB, Uni Würzburg], 4 SWS, 5 ECTS)

Possible electives (WS)

- Business Application Re-Engineering - Legacy Systems Architectures and Modernisation for Digital Transformation (Brune [FH Neu-Ulm], 4 ECTS, 5SWS, Virtuelle Hochschule Bayern [VHB]; please additionally submit a report about the course contents [10 pages])
- Introduction to Engineering Mathematics (Siller [Uni Würzburg], 5 ECTS, 4 SWS, Virtuelle Hochschule Bayern [VHB]; please additionally submit a report about the course contents [10 pages])
- Programming in C++ (Faber, 6 ECTS, 4 SWS, Virtuelle Hochschule Bayern [VHB])
- Special Devices and Circuits (Bogner, 5 ECTS, 4 SWS, ET/MT faculty; Thu, block3+4, E006)
- Quantum Computing (Glauner/Kunhardt, 4 SWS, 5 ECTS)
- Mobile and Wireless Networks (Kassler, 4 SWS, 5 ECTS)
- Technology Ethics and Sustainability (Valdes, please additionally submit a report about the course contents [10 pages], ITC2+ 0.27)

Teaching Methods

according to chosen elective

Remarks

according to chosen elective

Recommended Literature

according to chosen elective

MAI-11 FPGA PROGRAMMING

Module code	MAI-11
Module coordination	Prof. Thomas Limbrunner
Course number and name	MAI-11 FPGA Programmierung
Lecturer	Prof. Thomas Limbrunner
Semester	3
Duration of the module	1 semester
Module frequency	annually
Course type	required course
Level	Postgraduate
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	Portfolio
Weighting of the grade	5/90
Language of Instruction	English

Module Objective

Technical Competences

Students can implement small circuits using programmable hardware devices.

Students gain insight into hardware-level programming and design

Methodical Competences

Students are able to apply tools for FPGA programming and use them to implement a small project

Personal Competences

Students discover system architectures on a hardware-near level

Applicability in this and other Programs

-

Entrance Requirements

o Fundamental programming tools (e.g. control flow, data structures, functions)

- Digital logic (e.g., transistor, logic gate, K-map, SOP, multiplexer, counter,)
- Typical controller architecture (e.g., ALU, memory, peripherals,)

The course provides also an introduction to digital logic so you can still attend the course if you do not have any experience with digital logic.

Learning Content

- Typical structure of FPGAs
- Fundamentals about the FPGA tooling
- Introduction into VHDL
- Combinational logic
- Sequential logic
- State machines
- Memory
- Selected specific design aspects and application scenarios of FPGA technology

Teaching Methods

To reach the learning outcomes we will use the following didactic methods:

- Seminar-based teaching with many practical exercises (evaluation in a simulative environment as well as testing on target hardware)
- Self-reliant implementation of compact projects.
- Optional Flipped Classroom

Remarks

-

Recommended Literature

- Reichardt, Prof. Dr. J.; Schwarz, Prof. Dr. B.: VHDL-Simulation und -Synthese: Entwurf digitaler Schaltungen und Systeme, De Gruyter Studium, 8. Auflage, 978-3-110-67345-6
- Common literature on digital logic

MAI-12 COMPULSORY ELECTIVE SUBJECT OF A GENERAL ACADEMIC NATURE (AWP)

Module code	MAI-12
Module coordination	Tanja Mertadana
Course number and name	MAI-12-1 Compulsory Elective Subject of a General Academic Nature (AWP) I MAI-12-2 Compulsory Elective Subject of a General Academic Nature (AWP) II
Lecturer	Dozierende für AWP und Sprachen
Semester	1, 2
Duration of the module	2 semester
Module frequency	annually
Course type	required course
Level	Postgraduate
Semester periods per week (SWS)	4
ECTS	4
Workload	Time of attendance: 60 hours self-study: 60 hours Total: 120 hours
Type of Examination	See examination schedule AWP and languages
Weighting of the grade	4/90
Language of Instruction	Course dependent

Module Objective

This elective (AWP) course (i.e. compulsory elective subject of a general academic nature) enables students to acquire knowledge and skills in subject areas beyond the scope of their chosen degree programme.

Students can choose both face-to-face courses and courses offered by the Virtuelle Hochschule Bayern (VHB). Students acquire knowledge and skills in the following areas:

- in a foreign language (language skills)
- in the didactic-pedagogical area (methodological skills)
- in the social sciences (social skills)
- in the psychological-sociological field (social skills)
- in the technical and scientific field (professional skills)

o in the philosophical-social-ethical area (personal skills)

International students (without completed German B2 level) need to take a German course as part of elective (AWP) course I and II and receive ECTS starting from level German B1/ part 1 + 2. Native speakers of German or international students with German language skills of level C1 may choose any two elective (AWP) courses from the catalogue of the Language Centre.
As English B2 was an admission requirement, English can only be chosen at level C1.

Qualification objectives can be found in the corresponding course description on the homepage of the Language Centre:
https://th-deg.de/language-and-electives-centre

Applicability in this and other Programs

Applicable in other degree programmes.

Entrance Requirements

Language courses: Proof of successful completion of the previous language level must be provided.

Elective (AWP) courses may not have any overlapping content with the student's current degree programme.

Learning Content

The course content can be found in the corresponding course description on the homepage of the Language Centre:
https://th-deg.de/language-and-electives-centre

Teaching Methods

The teaching and learning methods can be found in the corresponding course description on the homepage of the Language Centre:
https://th-deg.de/language-and-electives-centre

Remarks

For course-specific details, please refer to the corresponding course description on the homepage of the Language Centre:
https://th-deg.de/language-and-electives-centre

All language courses require a compulsory attendance rate of 75% in order to be allowed to take the examination.

Recommended Literature

Recommended reading can be found in the corresponding course description on the homepage of the Language Centre:
https://th-deg.de/language-and-electives-centre

MAI-13 MASTERMODUL

Module code	MAI-13
Module coordination	Betreuer der Abschlussarbeit Supervisor of thesis
Course number and name	MAI-13 Master's Thesis MAI-13 Master's Colloquium
Lecturer	Prof. Dr. Peter Faber
Semester	3
Duration of the module	1 semester
Module frequency	each semester
Course type	required course
Level	Postgraduate
Semester periods per week (SWS)	2
ECTS	25
Workload	Time of attendance: 30 hours self-study: 720 hours Total: 750 hours
Type of Examination	oral ex. 30 min., master thesis
Weighting of the grade	25/90
Language of Instruction	English

Module Objective

-

Applicability in this and other Programs

not applicable

Entrance Requirements

according to study and examination regulation

Learning Content

individually depending on the topic

Teaching Methods

none

Remarks

-

Recommended Literature

individually depending on the topic

MAI-13 MASTER'S THESIS

Objectives

By producing a masters thesis, the students should demonstrate their ability to apply the knowledge and skills acquired during the M-AID curriculum, in an independently written scientific work on complex tasks. They thus demonstrate that they have successfully completed their master's levels studies and acquired the capacity for independent scientific work.

The students achieve the following learning objectives in the module:

Professional skills Students acquire the ability to immerse themselves in tasks of a scientific and technical nature and analyze and resolve problems on their own. They are able to tackle and solve even major tasks.

Methodological skills Using their scientific knowledge, students acquire the ability to tackle and resolve, unassisted, a large-scale issue of relevance to science in artificial intelligence and data science. The students deepen and apply the methods and instruments learned during their studies.

Social / Personal skills Students are able to tackle, independently and in application of self-discipline, a definable project of practical relevance to artificial intelligence and data science from a scientific perspective. The possibility of data collection and cooperation with companies opens up new experiences and career opportunities for students.

Type of Examination

master thesis

MAI-13 MASTER'S COLLOQUIUM

Objectives

A professional delivery of scientific and technical findings during the masters thesis, to be held as presentation, is integral to the successful completion of the master degree. This includes presenting results achieved and presenting complex linkages within a tight time frame.

Students will achieve the following learning objectives:

Professional skills
Students will be able to present the at times difficult technical and scientific relationships outlined in their masters thesis to an expert audience in the form of an oral presentation, and respond to questions about their presentation at an appropriate length.

Methodological skills
Students can intelligibly convey the nature and content of the findings from their masters thesis to an expert audience and present them within a defined time frame.

Social / Personal skills
Students are able to outline the outcomes in a presentation. The scenario of holding a presentation before an expert audience serves as a precursor to numerous similar situations students will encounter during their careers, especially with regard to time constraints and focusing on core messages; as such, this seminar prepares them for similar work-related situations.

Type of Examination

oral ex. 30 min.