TECHNISCHE HOCHSCHULE DEGGENDORF

Module Guide
Applied Computer Sciences

Faculty Computer Science
Examination regulations 07.12.2020
Date: 05.09.2024 11:35

Table of Contents

01 Theoretical Computer Science

Module code	01
Module coordination	Prof. Dr. Peter Faber
Course number and name	Computability Complexity Theory Formale Sprachen und Compilerbau I
Lecturers	Prof. Dr. Peter Faber Prof. Dr. Peter Jüttner
Semester	1
Duration of the module	1 semester
Module frequency	annually
Course type	required course
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.
Language of Instruction	English, German

Module Objective

The goal of this course is that students are able to understand and to apply formal theories and methods in the area 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 problem and application of resulting consequences for programming of software.

Methodical Competences

- Application of mathematical proof concepts

Computability Complexity Theory

Objectives

The goal of this course is that students are able to understand and to apply formal theories and methods in the area 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 problem and application of resulting consequences for programming of software.

Methodical Competences

- Application of mathematical proof concepts

Learning Content

Type of Examination

written ex. 90 min.

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 (availble also 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, Addison-Wesley, 2001
- Hopcroft, Motwani, Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson, 2002.

Formale Sprachen und Compilerbau I

Type of Examination

written ex. 90 min.

02 Practical Computer Science

Module code	02
Module coordination	Prof. Dr. Peter Jüttner
Course number and name	Formale Sprachen und Compilerbau II Advanced Software Engineering Programming Lab - Praktische Informatik Programmierpraktikum
Lecturers	Dr. Karsten Becker Prof. Dr. Peter Faber
Semester	1
Duration of the module	1 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	5/90
Language of Instruction	English, German

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

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 projecttypically 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.

Formale Sprachen und Compilerbau II

Type of Examination

written ex. 90 min.

Advanced Software Engineering

Type of Examination

written ex. 90 min.

Programming Lab - Praktische Informatik Programmierpraktikum

Type of Examination

written ex. 90 min.

03 Selected Topics of Embedded Software Development I

Module code	03
Module coordination	Prof. Dr. Andreas Grzemba
Course number and name	Embedded Connectivity Embedded Security
Lecturers	Prof. Dr. Andreas Grzemba Stefanie Merz
Semester	1
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	written student research project
Weighting of the grade	5/90
Language of Instruction	English

Module Objective

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

Applicability in this and other Programs

Elective subject in Master Electrical engineering and information technology

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

Lecture and Lab work

Recommended Literature

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

Embedded Connectivity

Type of Examination

student research project

Embedded Security

Type of Examination

student research project

04 Selected Topics of Embedded Software Development II

Module code	04
Module coordination	Prof. Dr. Christoph Schober
Course number and name	AI-M04 Selected Topics of Embedded Software Development II
Lecturer	Prof. Dr. Christoph Schober
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	written ex. 90 min.
Duration of Examination	90 min.
Weighting of the grade	according to ECTS
Language of Instruction	English

Module Objective

Knowledge and understanding
Students gain insights into a typical software development process in companies and large organizations. They will be able to work with project management tools such as JIRA to prioritize and plan development work. They know how to set up a modern software project with version control (Git) and continuous integration and continuous delivery pipelines (Gitlab CICD). They understand the importance of testing and code reviews for their professional work, but also in the context of certifications such as ISO 27001.

Application, utilisation and generation of knowledge
The tools and techniques used during the course are well-known and popular examples of their respective area. Students will be able to transfer their experience and knowledge to alternatives tools (for example from Gitlab to Github) and adapt to different work environments and toolstacks used throughout the software development industry.

Communication and Cooperation
Students learn to work together in a realistic team setup with different roles (such as software developer, product owner, test engineer and others). Different teams work together to build a working prototype, requiring extensive communication within and between the subteams to succeed. The students learn to discuss and resolve different opinions in regular Scrum-meetings.

Applicability in this and other Programs

This module can be used in other degrees.

Entrance Requirements

- Basic knowledge of programming
- Basic knowledge of software engineering

Learning Content

All practical work is done in the context of two competing start-ups ("Quantum Glow Inc." and "Colorbit GmbH") that aim to bring an innovative product to the market as soon as possible, using agile development techniques and modern project management. While the students will write code and work with hardware to really create such a product, the focus (and grading) is on the process, not the product.

- Software project management
    - Exemplary tool: Atlassian Jira for project management
    - Deep dive into Jira (project setup, features, how to use it)
- Agile software development
    - Exemplary technique: Scrum
    - Scrum meetings in theory and reality
    - Priorization techniques
    - the importance of retrospectives
- Version control
    - De-facto-standard: Git
    - Platforms with additional features and collaboration
    - Github
    - Gitlab

    - Sourceforge
    - Exemplary platform: Gitlab
    - Collaborative features
    - Security features
- Continous Integration and Delivery
    - Gitlab vs Github vs ???
    - Example: Gitlab CI/CD

Teaching Methods

lectures, practical project work, exercises

Recommended Literature

Online Resources

- Introduction to Git: https://git-scm.com/docs/gittutorial
- Introduction to Gitlab: https://docs.gitlab.com/ee/tutorials/
- Introduction to Gitlab CI/CD: https://docs.gitlab.com/ee/ci/

Books

- Scrum for dummies (ISBN 978-1-119-90467-0): https://ebookcentral.proquest.com/lib/th-deggendorf/detail.action?docID=7109023 (English)
- Scrum: kurz & gut (ISBN 9783868998337) (German)

05 Special Mathematical Methods

Module code	05
Module coordination	Prof. Dr. Thorsten Matje
Course number and name	Special Mathematical Methods
Lecturers	Prof. Dr. Peter Jüttner Prof. Dr. Thorsten Matje
Semester	2
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	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.

Applicability in this and other Programs

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

Entrance Requirements

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

2 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

3 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

4 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

5 Monte Carlo Simulation
- Business Planning Example
- Markov Chains and Metropolis?Hastings Algorithm

6 Fitting Data
- Least Squares Method
- Linear Least Squares
- Nonlinear Least Squares

Teaching Methods

Lectures and exercises

06 Elective Courses 1 - 5

Module code	06
Module coordination	Prof. Dr. Peter Jüttner
Course number and name	Electives 1 - 5
Lecturers	Prof. Dr. Peter Jüttner Dozierende der ausgewählten Wahlpflichtfächer Lecturer of the chosen Electives
Semester	1
Duration of the module	1 semester
Module frequency	annually
Course type	compulsory course
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
Language of Instruction	English

Module Objective

Electives 1 - 5

Type of Examination

Examination form of the chosen module

11 FPGA Programmierung

Module code	11
Module coordination	Gökçe Aydos
Course number and name	11 FPGA Programmierung
Lecturer	Gökçe Aydos
Semester	1
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	written ex. 90 min.
Duration of Examination	90 min.
Weighting of the grade	5/90
Language of Instruction	English

Module Objective

The purpose of the course is for you (the student) to learn to:

- explain the typical structure of FPGAs
- know fundamental FPGA tools
- apply basic features of SystemVerilog to describe
    - combinational logic
    - sequential logic
    - state machines
    - memory
    - a programmable processor: register file, program counter, ALU, instruction memory

    - bus
- apply various processor types to a given problem scenario
- know what lack of timing constraints can lead to
- use FPGA primitives to implement non-synthesizable features like clock synthesizer or analog-digital-converter
- understand how peripherals are connected to a microprocessor
- integrate logic that was implemented through high-level synthesis into a system-on-a-chip
- understand how an accelerator interacts with a microprocessor on a system-on-a-chip

Entrance Requirements

- Fundamental programming tools (e.g, control flow, data structures, functions)
- Digital logic (e.g., transistor, logic gate, K-map, SOP, POS, multiplexer, counter)
- Computer architecture (e.g., ALU, cache, memory, peripherals)

The learning materials contain a graceful introduction to digital logic so you can still attend the course if you do not have any experience with digital logic. But expect more workload in this case.

Learning Content

The purpose of the course is for you (the student) to learn to:

- explain the typical structure of FPGAs
- know fundamental FPGA tools
- apply basic features of SystemVerilog to describe
    - combinational logic
    - sequential logic
    - state machines
    - memory
    - a programmable processor: register file, program counter, ALU, instruction memory
    - bus
- apply various processor types to a given problem scenario
- know what lack of timing constraints can lead to
- use FPGA primitives to implement non-synthesizable features like clock synthesizer or analog-digital-converter
- understand how peripherals are connected to a microprocessor
- integrate logic that was implemented through high-level synthesis into a system-on-a-chip

- understand how an accelerator interacts with a microprocessor on a system-on-a-chip

Teaching Methods

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

- Flipped classroom
- Labs with feedback sessions

Recommended Literature

- Digital Logic
- FPGA Design for Embedded Systems Specialization Coursera

12 AWP

Module code	12
Module coordination	Tanja Mertadana
Course number and name	AWP I AWP II
Lecturer	Dozierende für AWP und Sprachen
Semester	1, 2
Duration of the module	2 semester
Module frequency	annually
Course type	compulsory elective 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	Prüfung Sprachenzentrum / AWP
Weighting of the grade	
Language of Instruction	German

Module Objective

Entrance Requirements

Learning Content

German for non-German students. Germans can choose a random language that is offered by the language centre.

Remarks

Duration of the module examination

- German exams (4 ECTS): 90 minutes
- all other language exams (2 ECTS): 60 minutes

Course language is the respectice foreign language.

AWP I

Type of Examination

See examination schedule AWP and languages

AWP II

Type of Examination

See examination schedule AWP and languages

13 Mastermodul

Module code	13
Module coordination	Prof. Dr. Peter Jüttner
Course number and name	Master's Thesis Master's Colloquium
Lecturers	Prof. Dr. A Admin Prof. Dr. Peter Faber Prof. Dr. Peter Jüttner N.N. Betreuer der Abschlussarbeit Supervisor of thesis
Semester	3
Duration of the module	1 semester
Module frequency	each semester
Course type	required course
Semester periods per week (SWS)	4
ECTS	23
Workload	Time of attendance: 60 hours self-study: 690 hours Total: 750 hours
Type of Examination	master thesis
Language of Instruction	German

Module Objective

Master's Thesis

Type of Examination

student research project

Master's Colloquium

Type of Examination

oral examination