DEGGENDORF INSTITUTE OF TECHNOLOGY

Module Handbook

Programme

Applied Computer Sciences
(Master)

Faculty

Faculty of Electrical Engineering, Media Technology and Computer Science

Examination Regulations

SS 2015

Contents

Introduction

This module handbook contains modules offered especially for the Master's Programme, Applied Computer Sciences. Modules offered in cooperation with the Electrical Engineering and Media Technology Master's programmes are described in the module handbooks of these programmes.

All courses are offered in German or in English when required.

Last Updated 21.05.15

0-01 Theoretical Informatics

Module No.	0-01
Module Head	Prof. Dr. Peter Jüttner
Course No. and Course Name	0-01-1 Semantics of Programming Languages, Computability and Complexity Theory 0-01-2 Formal Languages and Compiler Construction
Instructors	Prof. Dr. Peter Jüttner, Prof. Dr. Peter Fabers
Semester	1st
Length of Module	1 Semester
Module Frequency	annually
Status in Curriculum	Mandatory Subject
SWS (weekly semester hours)	6
ECTS	8
Workload	Class time: 0 hours Total: 0 hours
Language of Instruction	German /English

Module Objectives

The students should be in a position to recognise key elements of theoretical informatics and to apply the appropriate concepts and methods at a scientific level.

0-01-1 Semantics of Programming Languages, Computability and Complexity Theory

Course Objectives

The students should be in a position to apply key concepts and methods of theoretical informatics to scientific and technical tasks in both their studies and in a professional environment.

Furthermore, students should develop the following competencies: they are familiar with fundamental concepts of the semantics of programming languages in the form of the Fixed Point Theorem (the semantics of recursive functions), Operative Semantics (semantics based on machine-run programmes), and Axiomatic Semantics (mathematical semantics with the help of assertions). In the area of Computability, the students become familiar with various levels of computability, the existence of uncomputable problems and the related derivations and proofs. Complexity Theory will be introduced briefly.

Content

- Semantics
  - Definition, history
  - The semantics of recursive functions (Fixed Point Theorem)
  - Operative Semantics
  - Axiomatic Semantics
- Computability

  - Definition
  - Example of an uncomputable function
  - Turing machines and programming
  - LOOP, WHILE and GOTO computability
  - Computability and Recursion
- Complexity Theory
  - Definition
  - O-Notation
  - Complexity Levels

Admission Requirements and Recommended Prerequisites

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

Type of Examination

Written exam, 90 min. (module examination)

Methods

3 SWS (weekly semester hours) seminar-style lesson with exercises

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 available 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.

0-01-2 Compiler Construction

Course Objectives

This course introduces the fundamental theoretical principles of compiler construction. The students understand theoretical computer models and the structure of a compiler.

Content

- Introduction
- Translator
- Formal Languages
- Lexical Analysis
- Syntactic Analysis
- Syntax-based Translation
- Intermediate Code Generation
- Runtime Systems
- Code Generation
-

Admission Requirements and Recommended Prerequisites

-

Type of Examination

Written exam, 90 min. (module examination)

Methods

3 SWS seminar-style lesson

Literature

- Compilers – Principles, Techniques, and Tools. Aho, Lam, Sethi, Ullmann; Addison-Wesley :2nd Edition 2007
- Engineering a Compiler. Cooper, Torczon; 2nd Edition, Morgan Kaufmann: 2012
- Introduction to Automata Theory, Languages, and Computation. Hopcroft, Motwani, Ullman; Addison-Wesley: 2001
-

0-02 Practical Informatics

Module No.	0-02
Module Head	Prof. Dr. Peter Jüttner
Course No. and Course Name	0-02-1 Advanced Software Engineering 0-02-2 Compiler Construction Lab / Practical Training 0-02-3 Project
Instructors	Prof. Dr.Peter Jüttner, Prof. Dr. Peter Fabers
Semester	1st
Length of Module	1 Semester
Module Frequency	Annually
Status in Curriculum	Mandatory Subject
SWS	6
ECTS	8
Workload	In-class time: 0 hours Total: 0 hours
Language of Instruction	German / English

Module Objectives

The students should gain in-depth knowledge of selected topics in practical informatics and be able to apply the appropriate methods in practice in both scientific and industrial contexts.

0-02-1 Advanced Software Engineering

Objectives

This course presents in-depth knowledge of selected important topics in software engineering. The theory of these topics will be discussed and trained through practical examples.

Content

- □□□□□□□□□□□□
- Agile Methods - General
- Agile Methods - Scrum
- Important UML-Diagrammes in Detail
- Software Review Techniques
- Intensive Reviews

Admission Requirements and Recommended Prerequisites

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□

Type of Examination

Written exam, 90 Min. (Module examination)

Methods

2 SWS Seminar-style lesson with practical examples

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

0-02-2 Compiler Construction Lab / Practical Training

Objectives

This course presents in-depth knowledge of the practical side of compiler construction. The students implement a compiler in an ongoing practical exercise assigned by the instructor.

Content

- Introduction
- Translator
- Formal Languages
- Lexical Analysis
- Syntactic Analysis
- Syntax-based Translation
- Intermediate Code Generation
- Runtime Systems
- Code Generation
-

Admission Requirements and Recommended Prerequisites

-

Type of Examination

Written exam, 90 min. (Module examination)

Methods

3 SWS seminar-style lesson

Literature

- Compilers – Principles, Techniques, and Tools; Aho, Lam, Sethi, Ullmann; Addison-Wesley; 2nd Edition 2007

- Engineering a compiler; Cooper, Torczon; 2nd Edition, Morgan Kaufmann 2012
- Introduction to Automata Theory, Languages, and Comoputation; Hopcroft, Motwani, Ullman;Addison-Wesley 2001
- Further literature assigned by the instructor
- Introduction to Automata Theory, Languages, and Computation; Hopcroft, Motwani, Ullman; Addison-Wesley: 2001

0-02-3 Project

Content

The students work on a current programming task in teams in which they develop their own solutions to a given problem; as might occur in action in an actual software firm.

Admission Requirements and Recommended Prerequisites

Courses (Bachelor):

- Fundamentals of Informatics

- Introduction to Programming

- Software Engineering

Knowledge of programming and software development

Type of Examination

Written exam, 90 min. (Module examination)

Methods

The students analyse a problem assigned by the instructor, develop their own solutions and implement them.

Feedback sessions will be arranged with the instructor according to each individual assignment. Support through the E-Learning system.

Literature

Dependent on the particular assignment

0-03 Selected Topics in Embedded Software Development

Modul No.	0-03
Module Head	Prof. Dr. Peter Jüttner
Course No. and Course Name	0-03-1 Embedded Connectivity 0-03-2 Embedded Safety
Instructors	Prof. Dr. Terezia Toth, Prof. Dr. Andreas Grzemba
Semester	1st
Length of Module	1 Semester
Module Frequency	annually
Status in Curriculum	Mandatory subject
SWS	4
ECTS	5
Workload	In-class time: 0 hours Total: 0 hours
Language of Instruction	German / English

Module Objectives

The students should gain in-depth knowledge of selected topics in practical informatics and be able to apply the appropriate methods in practice in both scientific and industrial contexts.

0-03-1 Embedded Connectivity

Content

- Areas of application of industrial communication: Automation Technology and Automobile
- Types, basic principles and selection criteria for communication systems
- Possible applications in explosive / hazardous areas
- Design of communication systems
- Engineering und operation of communication systems
- Test and certification

Admission Requirements and Recommended Prerequisites

- none

Type of Examination

Written exam, 90 min. (Module examination)

Methods

- Seminar-style lesson with practical exercises, partly group work

Literature

- Tanenbaum, A., Wetherall, D.: Computer Networks, 5th Edition, 2011, Prentice Hall, ISBN 978-0-13-212695-3
- Sauter, M.: Grundkurs Mobile Kommunikationssysteme, 5. Auflage, 2013, Vieweg, ISBN 978-3-8348-1407-4
- Matheus, K., Königseder, T.: Automotive Ethernet, 2014, Cambridge University Press, ISBN 978-1107057289

0-03-2 Embedded Safety

Content

- tbd.

Admission Requirements and Recommended Prerequisites

- tbd.

Type of Examination

Written exam, 90 min. (Module examination)

Methods

- Seminar-style lesson with practical exercises, partly group work

Literature

- tbd.

0-11 FPGA Programming

Module No.	0-011
Module Head	Prof. Dr. Peter Jüttner
Course No. and Course Name	0-011 FPGA Programming
Instructors	Martin Schramm (lecturer)
Semester	1st
Length of Module	1 Semester
Module Frequency	annually
Status in Curriculum	Mandatory subject
SWS	4
ECTS	5
Workload	In-class time: 0 Hours Total: 0 Hours
Language of Instruction	German /English

Objectives

The students become familiar with the key principles of FPGA hardware design by means of VHDL on a theoretical level as well as through practical examples and are able to apply them in both a professional and academic environment.

Content

- Introduction and Motivation
- Modelling Digital Systems with VHDL
    - Basic concepts of VHDL
    - Behavioural and structural description
    - Type concept
    - Sequential and parallel statements
    - Procedures and functions
- Realisation of Digital Circuits
- Methods of Hardware Debugging
    - Netlist analysis
    - Simulation of a digital design system
    - Logic analysis by means of a virtual logic analyser
- System design

Admission Requirements and Recommended Prerequisites

Lectures:

- Fundamentals of Informatics
- Introduction to Programming
- Digital Technology

- Computer Networks

- System Programming

Type of Examination

Written exam, 90 min.

Methods

Seminar-style lesson with practical exercises, partly group work

Literature

- J. Reichardt, B. Schwarz, VHDL Synthese: Entwurf digitaler Schaltungen und Systeme, Oldenbourg Wissenschaftsverlag

- J. Ritter, P. Molitor, VHDL: Eine Einführung, Addison-Wesley Verlag