inf030 - Programming, Algorithms and Data Structures .......................................................... 4
inf031 - Object-oriented Modelling and Programming .......................................................... 7
inf200 - Computer Engineering I ......................................................................................... 10
inf700 - Computer Science Education I ............................................................................. 12
mat950 - Discrete Mathematics .......................................................................................... 14
inf005 - Software Engineering I .......................................................................................... 16
inf007 - Information Systems I ............................................................................................ 18
inf700 - Computer Science Education I ............................................................................. 20
inf004 - Software Project .................................................................................................... 22
inf009 - Database Practical ................................................................................................... 25
inf014 - Operating Systems Practical .................................................................................. 27
inf018 - Media Processing .................................................................................................... 29
inf021 - Advanced Java Technologies .................................................................................. 32
inf202 - Computer Engineering Practical ........................................................................... 34
inf800 - Proseminar in Computer Science .......................................................................... 37
inf803 - Special Topics in Computer Science I .................................................................... 39
inf804 - Special Topics in Computer Science II .................................................................. 41
inf808 - Current Topics in Computer Science ...................................................................... 43
inf200 - Computer Engineering I ......................................................................................... 45
inf201 - Computer Engineering II ....................................................................................... 47
inf203 - Embedded Systems I .............................................................................................. 49
Modules for Computing Science

Basismodule

inf030 - Programming, Algorithms and Data Structures

Module label: Programming, Algorithms and Data Structures
Module abbreviation: inf030
Credit points: 9.0 KP
Workload: 270 h

Applicability of the module:
- Bachelor's Programme Business Informatics (Bachelor) > Basiscurriculum
- Bachelor's Programme Computing Science (Bachelor) > Basismodule
- Bachelor's Programme Economics and Business Administration (Bachelor) > Studienrichtung Wirtschaftsinformatik
- Bachelor's Programme Mathematics (Bachelor) > Nebenfachmodule
- Bachelor's Programme Sustainability Economics (Bachelor) > Wahlpflichtbereich
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule

Responsible persons:
- Schönberg, Christian (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

Prerequisites: No participant requirement

Skills to be acquired in this module:
- Programming is one of the basic activities of computer scientists and a prerequisite for many other courses in computer science studies. The aim of the module "Programming, Data Structures and Algorithms" is to learn the basic concepts of imperative, procedural and object-oriented programming using the Java programming language and to present known, efficient algorithms and data structures for various, frequently occurring problems.
- After completing the module, the students should be able to independently develop imperative and simple object-oriented programmes based on Java for solving smaller problems and assess the efficiency of their programmes. They should also be able to apply important algorithms and select them based on their complexity.

Professional competence:
The students:
- describe basic concepts of imperative programming with Java
- recognise imperative programming terminology and use the appropriate terms accurately in discussions
- recognise basic terminology of object-oriented programming
- describe what programs presented to them do
- independently develop programs to solve small problems
- systematically examine their own and other people's programmes for errors
- use modern programme development environments to develop and test programmes
- create algorithms with general design concepts (e.g. Greedy method, divide-and-conquer method)
- name algorithms and data structures for solving common problems and evaluate their applicability
- name problems of efficiency of algorithmic solutions of concrete problems and evaluate them
- make a well-founded choice of an algorithm and a data structure for solving a concrete problem
- apply the learned algorithms and data structures sensibly to given and concrete problems

Methodological competence:
The students:
- solve given problems from the point of view of imperative or object-oriented programming
- transfer practical experience in programme development to new tasks

Social competence:
The students:
• communicate the structure and mode of operation of self-developed
  programmes to others
• present solutions to small tasks in front of groups

Self-competence

The students:

• organise themselves in finding algorithmic solutions to small and
  medium-sized problems in computer science
• incorporate the concepts of general programme design in their actions

Module contents

In the first part, general basic concepts of programming are introduced:

• Algorithm, programming languages, computer
• development tools, development phases
• compilers
• grammars
• logic

The second part deals with basic programming concepts:

• data types
• variables
• Expressions, statements
• control structures
• Methods, parameters
• recursion
• Reference data types, arrays
• Classes, objects
• Documentation
• Testing

The third part contains an introduction to data structures and algorithms as well
as a discussion of their efficiency, i.e. the computational effort or memory
requirements depending on the amount of data to be processed. The module
introduces known, efficient algorithms and data structures for various,
frequently occurring problems. These include in particular:

• Methods for searching for keys, as well as insertion and deletion
  in dynamic data sets, e.g. lists, trees, AVL trees or hash methods,
• Methods for searching for text patterns,
• Methods for sorting data by key values, e.g. QuickSort and HeapSort,
• Graph-based applications, e.g. for finding shortest paths in graphs.

The lecture part is supplemented by a comprehensive exercise part, in
which in particular the taught programming contents are implemented in
practical examples.

Recommended reading

Essential:

• Lecture Notes (made available either in printed form or via Stud.IP
during the course of the lecture)

Good secondary literature:

• Dietmar Ratz, Jens Scheffler, Detlev Seese, Jan Wiesenberger:
  Grundkurs Programmieren in Java, Carl Hanser Verlag.
• Joachim Goll, Cornelia Heinisch: Java als erste Programmiersprache,
  Springer Vieweg Verlag
• Ottmann, Widmayer: Algorithmen und Datenstrukturen. Spektrum
  Verlag, 5. Auflage, 2012
• Siege: Einführung in die Informatik. Shaker Verlag, 2013
### Language of instruction
German

### Duration (semesters)
1 Semester

### Module frequency
every winter term

### Module capacity
unlimited

### Module level

### Type of module

<table>
<thead>
<tr>
<th>Teaching/Learning method</th>
<th>1V + 1Ü</th>
</tr>
</thead>
</table>

### Previous knowledge
none

### Examination

<table>
<thead>
<tr>
<th>Examination</th>
<th>Examination times</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td>At the End of the Semester</td>
<td>Portfolio or written exam or oral exam</td>
</tr>
</tbody>
</table>

### Workload of compulsory attendance

<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>4</td>
<td>WiSe</td>
<td>56</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>2</td>
<td>WiSe</td>
<td>28</td>
</tr>
</tbody>
</table>

### Total module attendance time
84 h
inf031 - Object-oriented Modelling and Programming

<table>
<thead>
<tr>
<th>Module label</th>
<th>Object-oriented Modelling and Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf031</td>
</tr>
<tr>
<td>Credit points</td>
<td>9.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>270 h</td>
</tr>
</tbody>
</table>
| Applicability of the module         | • Bachelor's Programme Business Informatics (Bachelor) > Basiscurriculum  
• Bachelor's Programme Computing Science (Bachelor) > Basismodule  
• Bachelor's Programme Economics and Business Administration (Bachelor) > Studiennrichtung Wirtschaftsinformatik  
• Bachelor's Programme Mathematics (Bachelor) > Nebenfachmodule  
• Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule |
| Responsible persons                 | Schönberg, Christian (module responsibility)  
Lehrenden, Die im Modul (authorised to take exams) |
| Prerequisites                       | useful previous knowledge: inf030 Programming, Algorithms and Data Structures |
| Skills to be acquired in this module| Object orientation represents the state of the art in software development today. Given problems are first transformed into an object-oriented model and then into an object-oriented program with the help of object-oriented analysis and design methods. Aim of the module "Object-Oriented Modelling and Programming" is to learn basic concepts of object-oriented modelling using UML as the modelling notation and object-oriented programming using the Java programming language. After completing the module, students should be able to independently develop object-oriented programs based on Java for solving medium-sized problems.  
**Professional competence:**  
The students:  
• know basic concepts of object-oriented modelling and UML as modelling notation  
• know basic concepts of object-oriented programming with Java  
• know the terminology of object-oriented modelling and programming and use the appropriate terms precisely in discussions  
• can describe what object-oriented programmes presented to them do  
• independently develop models and programmes for solving medium-sized problems  
• systematically examine their own and other people's models and programmes for errors  
• use modern development environments for modelling and developing programmes  
• know the differences between the imperative, object-oriented, functional, logical and rule-based programming paradigms  
**Methodological competence:**  
The students:  
• independently develop programmes for given problems by consistently applying the concepts of object-oriented modelling and programming  
• transfer practical experience in programme development to new tasks  
• independently develop programmes with concurrency  
• can independently apply known solution methods to complex problems  
**Social competence:**  
The students:  
• communicate the structure and mode of action of self-developed models and programmes to others  
• present independently developed solutions to groups  
**Self-competence:**  
The students:  
• organise themselves when developing programmes for small and medium-sized problems in computer science  
• incorporate the concepts of object-oriented programme design in their actions |
Module contents

In the first part, basic concepts of object-oriented modelling and programming are taught:

- Models and modelling
- UML class diagrams
- Classes and objects
- Data encapsulation
- Inheritance
- Polymorphism and dynamic binding
- Exception handling
- Genericity

In the second part, important concepts and classes of the JDK class library are introduced and the classes are used in solving medium-sized problems:

- Java Collection API
- IO and Streams
- GUI applications with JavaFX
- Parallel programming with threads

In the third part, advanced solution strategies are presented and other programming paradigms are introduced and compared with the object-oriented paradigm:

- Backtracking, Branch and Bound, Greedy
- Local Search, Evolutionary Algorithms
- Functional programming (e.g. Java-Lamdas, Standard ML)
- Logical programming (e.g. Prolog)
- Rule-based programming (e.g. Drools)

The lecture part is supplemented by a comprehensive exercise part, in which in particular the taught contents are implemented in practical examples.

Recommended reading

Essential:

- Lecture Notes (made available either in printed form or via Stud.IP during the course of the lecture)

Good secondary literature:

- Heide Balzert: Lehrbuch der Objektmodellierung: Analyse und Entwurf mit der UML 2, Spektrum Akademischer Verlag
- Dietmar Ratz, Jens Scheffler, Detlev Seese, Jan Wiesenberger: Grundkurs Programmieren in Java, Carl Hanser Verlag.
- Christian Ullenboom: Java ist auch eine Insel: Programmieren lernen mit dem Standardwerk für Java- Entwickler, Rheinwerk Computing
- Christian Ullenboom: Java SE 8 Standard-Bibliothek: Das Handbuch für Entwickler, Rheinwerk Computing

Links

Language of instruction

German

Duration (semesters)

1 Semester

Module frequency

every summer term

Module capacity

unlimited

Module level

Type of module

Teaching/Learning method

1V + 1Ü

Previous knowledge

useful previous knowledge: inf030 Programming, Algorithms and Data Structures

Examination

Examination times

Type of examination

Final exam of module

At the end of the Semester.

Portfolio or written exam or oral exam
<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>4</td>
<td>SoSe</td>
<td>56</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>2</td>
<td>SoSe</td>
<td>28</td>
</tr>
<tr>
<td><strong>Total module attendance time</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>84 h</strong></td>
</tr>
</tbody>
</table>
inf200 - Computer Engineering I

Module label
Computer Engineering I

Module abbreviation
inf200

Credit points
6.0 KP

Workload
180 h

Applicability of the module
- Bachelor's Programme Computing Science (Bachelor) > Basismodule
- Bachelor's Programme Mathematics (Bachelor) > Nebenfachmodule
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Akzentsetzungsbereich

Responsible persons
Rauh, Andreas (module responsibility)
Lehrenden, Die im Modul (authorised to take exams)

Prerequisites
No participant requirements

Skills to be acquired in this module

The students learn to understand the construction of digital circuits and digital computers. They know the technological parameters, the state of the art technologies, and the developments characterizing current and future design paradigms for digital hardware. They learn to understand the concepts underlying current computer architectures and are able to explain how such architectures execute programs. Successful participants will be able to analyze computer architectures as a whole, to understand in depth, to analyze, and to optimize their hardware components, and to discuss the properties induced by selecting design alternatives.

Professional competences
The Students:
- identify fundamental concepts of the construction of digital computer systems, the internal number representation, and analysis of combinational logic as well as their optimization.

Methodological competences
The Students:
- analyze computer architectures on the basis of their individual components
- design and optimize digital hardware components
- transfer systematic approaches of hardware design to unknown design problems

Social competences
The Students:
- present their understanding of the functional principles of digital computer systems

Self-competences
The Students:
- critically reflect on the results of exercises and recognize limitations of different approaches to the design of digital computer systems

Module contents
This module is the first part of the introduction to computer engineering. It explains the construction principles of computers, from the implementation of an easy Instruction Set Architecture, over fundamental techniques for coding and representation of numbers, program execution on machine level, basics of logics and analysis of functions of combinational logic as well as their optimization.

Recommended reading
- Lecture Notes

<table>
<thead>
<tr>
<th>Links</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Language of instruction</td>
<td>German</td>
</tr>
<tr>
<td>Duration (semesters)</td>
<td>1 Semester</td>
</tr>
<tr>
<td>Module frequency</td>
<td>annual</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Module level</td>
<td></td>
</tr>
<tr>
<td>Type of module</td>
<td></td>
</tr>
<tr>
<td>Teaching/Learning method</td>
<td>1VL + 1Ü</td>
</tr>
<tr>
<td>Previous knowledge</td>
<td>none</td>
</tr>
<tr>
<td>Examination</td>
<td>Examination times</td>
</tr>
<tr>
<td>Final exam of module</td>
<td>At the end of the lecture term</td>
</tr>
<tr>
<td>Type of course</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>Lecture</td>
<td>Comment SWS Frequency Workload of compulsory attendance</td>
</tr>
<tr>
<td>Exercises</td>
<td>Comment SWS Frequency Workload of compulsory attendance</td>
</tr>
<tr>
<td>Total module attendance time</td>
<td>56 h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3</td>
<td>WiSe</td>
<td>42</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1</td>
<td>WiSe</td>
<td>14</td>
</tr>
</tbody>
</table>

11 / 74
inf700 - Computer Science Education I

Module label: Computer Science Education I
Module abbreviation: inf700
Credit points: 6.0 KP
Workload: 180 h

Applicability of the module:
- Bachelor's Programme Business Informatics (Bachelor) > Wahlbereich Informatik, Kultur und Gesellschaft
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Bachelor's Programme Computing Science (Bachelor) > Wahlbereich Informatik, Kultur und Gesellschaft
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule (60 KP)
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule

Responsible persons:
- Diethelm, Ira (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

Prerequisites:
- Basic knowledge of computer science

Skills to be acquired in this module:

Professional competence
The students:
- characterise the different computer science education (CSE) concepts and approaches, e.g. the early approaches of CSE in school or the concept of computer science (CS) in contexts
- select and discuss teaching subjects by analysing didactic approaches and concepts
- describe the general education character of CS
- compare the different approaches and concepts of CSE and are able to illustrate common features and contradictions
- reflect lesson subjects by the approaches and topics of CSE

Methodological competence
The students:
- link the concepts and approaches of CSE with the educational reconstruction
- classify the similarities and differences of the concepts and approaches of CSE academically

Social competence
The students:
- discuss the concepts and approaches of CSE with students and lectures academically
- accept the thoughts of other students and lectures
- give and accept criticism objectively

Self-competence
The students:
- integrate the concepts and approaches of CSE into their planning and operations - reflect their self-perception with regard to the concepts and approaches of CSE

Module contents
The field of CSE is introduced by this module. Different CSE approaches and concepts are presented. These CSE approaches and concepts are, e.g.:
- early concepts of CS in schools
- general education character of CS
- idea oriented approach of CSE
- information centred approach of CSE
- CSE in elementary school
- system oriented approach Subjects like „CS projects in class“ are also part of this module.

Recommended reading
**Links**

**Language of instruction**  German

**Duration (semesters)**  1 Semester

**Module frequency**  annual

**Module capacity**  unlimited

**Module level**

**Type of module**

**Teaching/Learning method**  1VL + 1 Ü

**Previous knowledge**  Basic knowledge of computer science

**Examination**

<table>
<thead>
<tr>
<th>Examination times</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>At the end of the lecture period</td>
<td>Oral exam</td>
</tr>
</tbody>
</table>

**Final exam of module**

<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>2</td>
<td>SoSe</td>
<td>28</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>2</td>
<td>SoSe</td>
<td>28</td>
</tr>
</tbody>
</table>

**Total module attendance time**  56 h
**mat950 - Discrete Mathematics**

**Module label**
Discrete Mathematics

**Module abbreviation**
mat950

**Credit points**
6.0 KP

**Workload**
180 h

**Applicability of the module**
- Bachelor's Programme Business Informatics (Bachelor) > Aufbaucurriculum - Pflichtbereich
- Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule

**Responsible persons**
- Heß, Florian (module responsibility)
- Stein, Andreas (module responsibility)
- Stein, Sandra (module responsibility)

**Prerequisites**
- Getting to know and to understand the axiomatic structure of mathematics and the importance of mathematical reasoning
- Mastering basic mathematical proof techniques and their logical structure
- Recognizing the relevance of premises in mathematical theorems: Localization of premises within proofs and possible consequences if premises are not met
- Exemplary acquaintance with further mathematical areas and thus expansion of the student's mathematical knowledge
- Getting to know applications
- Integration and crosslinking of the student's mathematical knowledge by establishing relationships between different mathematical areas
- Learning the essential ideas and methods for discrete structures in mathematics
- Knowledge of the fundamental concepts and methods of algebra and number theory, such as groups, rings, fields, residue class rings, Euclidean algorithm, Chinese remainder theorem, polynomials.
- Knowledge of further concepts and methods for discrete structures, e.g. primality tests, RSA, graph-theoretical algorithms

**Module contents**
- Elements of propositional logic, proof techniques, sets, relations and maps, combinatorics, graphs and applications, the ring of integers and residue class rings, groups and semi groups

**Recommended reading**
- Graham, Knuth, Patashnik: Concrete Mathematics, Addison-Wesley 1994.
- Hartmann: Mathematik für Informatiker, Vieweg 2014.
- Teschl, Teschl: Mathematik für Informatiker, Band 1, Springer 2013.

Further reading will be announced in the lecture.

**Links**

**Language of instruction**
German

**Duration (semesters)***
1 Semester

**Module frequency**
annual

**Module capacity**
unlimited

**Reference text**
Im Zwei-Fächer Bachelor Informatik ist dieses Modul im Basiscurriculum zu studieren.

**Module level**

**Type of module**

**Previous knowledge**

<table>
<thead>
<tr>
<th>Examination</th>
<th>Examination times</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td>after the end of the lecture period</td>
<td>Written exam or oral exam.</td>
</tr>
</tbody>
</table>

Bonus points can be earned.
<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3</td>
<td>WiSe</td>
<td>42</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1</td>
<td>WiSe</td>
<td>14</td>
</tr>
<tr>
<td><strong>Total module attendance time</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>56 h</strong></td>
</tr>
</tbody>
</table>
**Aufbaumodule (60 KP)**

**inf005 - Software Engineering I**

<table>
<thead>
<tr>
<th>Module label</th>
<th>Software Engineering I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf005</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
</tbody>
</table>

**Applicability of the module**

- Bachelor's Programme Business Informatics (Bachelor) > Aufbaucurriculum - Pflichtbereich
- Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule
- Bachelor's Programme Mathematics (Bachelor) > Nebenfachmodule
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule (60 KP)
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Pflichtbereich
- Master's Programme Environmental Modelling (Master) > Mastermodule

**Responsible persons**

- Winter, Andreas (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

**Prerequisites**

- inf030
- inf031

**Skills to be acquired in this module**

The objective of the module is to convey the development and maintainance of large scale software systems. The complete software developing process including requirements collection, software architecture and quality control is observed. The basics of object oriented modelling and software development are enhanced.

**Professional competence**

The students:

- comprehend the different developmental phases of software (especially requirements engineering, software design, software implementation and quality control)
- name the tasks of each phase
- select appropriate methodical utilities
- select suitable methods and utilities for each project phase
- understand the advantages of the modelling process with UML
- model moderate tasks in UML
- understand and develop solutions for given problems by means of development environments

**Methodological competence**

The students:

- structure, document and evaluate problems and solutions with the tools of object oriented modelling
- apply methods and techniques of object oriented modelling purposefully

**Social competence**

The students:

- create, present and discuss solutions with modelling techniques
- present and solve modelling problems in teams

**Self-competence**

The students: reflect their problem-solving behaviour with regard to the capabilities of software technology

**Module contents**

The module introduces fundamental terms and concepts in software engineering.

This includes:

- need for software engineering
- activities and process-models in software development
- object-oriented modelling, meta modelling
- Interdependencies between code and models
- requirements elicitation
- definition of software architectures
- application of software patterns
- software quality management
- software maintenance, evolution and operation Software engineering tools are presented and applied in practical exercises.

**Recommended reading**

- Slide script for the lecture
- Helmut Balzert: Lehrbuch der Software-Technik, Spektrum Akademischer Verlag, 3. Auflage 2009

**Links**

<table>
<thead>
<tr>
<th>Language of instruction</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration (semesters)</td>
<td>1 Semester</td>
</tr>
<tr>
<td>Module frequency</td>
<td>annual</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Module level</td>
<td></td>
</tr>
<tr>
<td>Type of module</td>
<td>1VL + 1Ü</td>
</tr>
<tr>
<td>Previous knowledge</td>
<td>inf030</td>
</tr>
<tr>
<td></td>
<td>inf031</td>
</tr>
<tr>
<td>Examination</td>
<td></td>
</tr>
<tr>
<td>Examination times</td>
<td></td>
</tr>
<tr>
<td>Type of examination</td>
<td></td>
</tr>
<tr>
<td>Final exam of module</td>
<td>At the end of the lecture period or during the lecture period (portfolio)</td>
</tr>
<tr>
<td></td>
<td>Written exam or oral exam or portfolio (3 services)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3</td>
<td>WiSe</td>
<td>42</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>2</td>
<td>WiSe</td>
<td>28</td>
</tr>
<tr>
<td>Total module attendance time</td>
<td></td>
<td>70 h</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### inf007 - Information Systems I

<table>
<thead>
<tr>
<th>Module label</th>
<th>Information Systems I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf007</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
</tbody>
</table>

#### Applicability of the module
- Bachelor's Programme Business Informatics (Bachelor) > Aufbaucurriculum - Pflichtbereich
- Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule
- Bachelor's Programme Economics and Business Administration (Bachelor) > Studienrichtung Wirtschaftsinformatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule (60 KP)
- Master Applied Economics and Data Science (Master) > Specialization
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Pflichtbereich

#### Responsible persons
- Wingerath, Wolfram (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

#### Prerequisites
No participant requirement

#### Skills to be acquired in this module
This module introduces the core concepts, languages and architectures of databases. In software systems these concepts are important.

**Professional competence**
The students:
- name the core concepts of the languages and architectures of databases (especially)
- select data models
- integrate structuring concepts of information systems in their designs

**Methodological competence**
The students:
- design database systems appropriately
- analyse problems from the field of database-supported information systems and solve them appropriately

**Social competence**
The students:
- enhance their ability to work in a team

**Self-competence**
The students:
- reflect their problem-solving behaviour with regard to the information processing concepts

#### Module contents
- Relational data models
- Relational algebras and its implementation in SQL (the standard of databases)
- Database design on different abstractions (conceptual and logical design)
- Normalisation - Data base architectures
- Distributed and active databases
- Object-oriented, object-related and XML-based database systems

#### Recommended reading

#### Language of instruction
German
<table>
<thead>
<tr>
<th><strong>Duration (semesters)</strong></th>
<th>1 Semester</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module frequency</strong></td>
<td>annual</td>
</tr>
<tr>
<td><strong>Module capacity</strong></td>
<td>unlimited</td>
</tr>
<tr>
<td><strong>Type of module</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Teaching/Learning method</strong></td>
<td>1VL + 1Ü</td>
</tr>
<tr>
<td><strong>Previous knowledge</strong></td>
<td>none</td>
</tr>
<tr>
<td><strong>Examination</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Examination times</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Type of examination</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Final exam of module</strong></td>
<td>At the end of the lecture period</td>
</tr>
<tr>
<td><strong>Written or oral exam</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Type of course</strong></th>
<th><strong>Comment</strong></th>
<th><strong>SWS</strong></th>
<th><strong>Frequency</strong></th>
<th><strong>Workload of compulsory attendance</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3</td>
<td>WiSe</td>
<td>42</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1</td>
<td>WiSe</td>
<td>14</td>
</tr>
<tr>
<td><strong>Total module attendance time</strong></td>
<td></td>
<td></td>
<td></td>
<td>56 h</td>
</tr>
</tbody>
</table>
**inf700 - Computer Science Education I**

<table>
<thead>
<tr>
<th>Module label</th>
<th>Computer Science Education I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf700</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
</tbody>
</table>

**Applicability of the module**
- Bachelor's Programme Business Informatics (Bachelor) > Wahlbereich Informatik, Kultur und Gesellschaft
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Bachelor's Programme Computing Science (Bachelor) > Wahlbereich Informatik, Kultur und Gesellschaft
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule (60 KP)
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule

**Responsible persons**
- Diethelm, Ira (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

**Prerequisites**
- Basic knowledge of computer science

**Skills to be acquired in this module**

**Professional competence**
The students:
- characterise the different computer science education (CSE) concepts and approaches, e.g. the early approaches of CSE in school or the concept of computer science (CS) in contexts
- select and discuss teaching subjects by analysing didactic approaches and concepts
- describe the general education character of CS
- compare the different approaches and concepts of CSE and are able to illustrate common features and contradictions
- reflect lesson subjects by the approaches and topics of CSE

**Methodological competence**
The students:
- link the concepts and approaches of CSE with the educational reconstruction
- classify the similarities and differences of the concepts and approaches of CSE academically

**Social competence**
The students:
- discuss the concepts and approaches of CSE with students and lectures academically
- accept the thoughts of other students and lectures
- give and accept criticism objectively

**Self-competence**
The students:
- integrate the concepts and approaches of CSE into their planning and operations - reflect their self-perception with regard to the concepts and approaches of CSE

**Module contents**
The field of CSE is introduced by this module. Different CSE approaches and concepts are presented. **These CSE approaches and concepts are, e.g.:**
- early concepts of CS in schools
- general education character of CS
- idea oriented approach of CSE
- information centred approach of CSE
- CSE in elementary school
- system oriented approach Subjects like „CS projects in class“ are also part of this module.

**Recommended reading**
<table>
<thead>
<tr>
<th>Links</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Language of instruction</td>
<td>German</td>
</tr>
<tr>
<td>Duration (semesters)</td>
<td>1 Semester</td>
</tr>
<tr>
<td>Module frequency</td>
<td>annual</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Module level</td>
<td></td>
</tr>
<tr>
<td>Type of module</td>
<td></td>
</tr>
<tr>
<td>Teaching/Learning method</td>
<td>1VL + 1 Ü</td>
</tr>
<tr>
<td>Previous knowledge</td>
<td>Basic knowledge of computer science</td>
</tr>
<tr>
<td>Examination</td>
<td></td>
</tr>
<tr>
<td>Examination times</td>
<td></td>
</tr>
<tr>
<td>Type of examination</td>
<td></td>
</tr>
<tr>
<td>Final exam of module</td>
<td>At the end of the lecture period</td>
</tr>
<tr>
<td></td>
<td>Oral exam</td>
</tr>
<tr>
<td>Type of course</td>
<td>Comment</td>
</tr>
<tr>
<td></td>
<td>SWS</td>
</tr>
<tr>
<td></td>
<td>Frequency</td>
</tr>
<tr>
<td></td>
<td>Workload of compulsory attendance</td>
</tr>
<tr>
<td>Lecture</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>SoSe</td>
</tr>
<tr>
<td></td>
<td>28</td>
</tr>
<tr>
<td>Exercises</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>SoSe</td>
</tr>
<tr>
<td></td>
<td>28</td>
</tr>
<tr>
<td>Total module attendance time</td>
<td>56 h</td>
</tr>
</tbody>
</table>
### Praktische Vertiefung (60 KP)

**inf004 - Software Project**

<table>
<thead>
<tr>
<th>Module label</th>
<th>Software Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module label</td>
<td>inf004</td>
</tr>
<tr>
<td>Credit points</td>
<td>9.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>270 h</td>
</tr>
</tbody>
</table>

**Applicability of the module**

- Bachelor's Programme Biology (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Business Administration and Law (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Business Informatics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Chemistry (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Comparative and European Law (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Computing Science (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Economics and Business Administration (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Education (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Engineering Physics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Environmental Science (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Intercultural Education and Counselling (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Mathematics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Physics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Physics, Engineering and Medicine (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Social Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Sustainability Economics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Art and Media (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Biology (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Chemistry (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Computer Science (Bachelor) > Praktische Vertiefung (60 KP)
- Dual-Subject Bachelor's Programme Economics and Business Administration (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Dutch Linguistics and Literary Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Economic Education (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Economics and Business Administration (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Education (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Elementary Mathematics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme English Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Gender Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme General Education (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme German Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme History (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-subject bachelor's programme Low German (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Material Culture: Textiles (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Mathematics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Music (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Philosophy / Values and Norms (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Physics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Protestant Theology and Religious Education (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Slavic Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Social Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Sport Science (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Technology (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel

Responsible persons
- Grawunder, Marco (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

Prerequisites
Java Programming Course, Software Engineering, Algorithms and Data Structures

Skills to be acquired in this module
The students will be able to develop software iteratively in a team. This includes all stages of the software life cycle (requirements, analysis, design, implementation, test) and the presentation of the software development process. The students improve their Java skills. Professional competence: The students: - Apply software development techniques and methods and are aware of the techniques' limitations

Professional competence:
The students:
- apply techniques and methods and recognize their limitations

Methodological competence
The students:
- develop complex software with software engineering methods using a process-model and document these appropriately
- make a rough schedule/estimate on tasks
- implement an iterative process
- familiarise themselves with unknown systems and frameworks
- process complex tasks based on science and engineering and split them in subtasks
- organise and implement small-scale projects
- present and document the outcome of the project

Social competence
The students:
- work in a team and solve conflicts
- develop complex software in a team and assess the required efforts (time management)
- reflect their self-performance and the performance of other students (review and retrospective)

Self-competence
The students:
• improve their capacity for teamwork, in particular the ability to solve conflicts

**Module contents**

In a two-semester course a team of students develops a larger system. For this project a Scrum similar model is used. Typical external stakeholders/roles are represented by members of the team. Feedback is provided in regular presentations (two per semester with the lecturer, weekly in the seminars). An accompanying lecture block provides the most important software engineering subjects and repeats or deepens new methods and techniques necessary for the project.

**Recommended reading**

- [https://confluence.swp.offis.uni-oldenburg.de/display/SWPWP](https://confluence.swp.offis.uni-oldenburg.de/display/SWPWP)

**Language of instruction**

German

**Duration (semesters)**

2 Semester

**Module frequency**

annual

**Module capacity**

unlimited

**Module level**

**Teaching/Learning method**

1VL + 1Ü + 1PR

**Previous knowledge**

Java Programming Course, Software Engineering, Algorithms and Data Structures

**Examination**

- **Final exam of module**
  - Examination times: by appointment
  - Type of examination: Portfolio

**Type of course**

- **Lecture**
  - Comment: SWS 2
  - Frequency: WiSe
  - Workload of compulsory attendance: 28
- **Exercises**
  - Comment: SWS 2
  - Frequency: SoSe und WiSe
  - Workload of compulsory attendance: 28
- **Project**
  - Comment: SWS 4
  - Frequency: SoSe und WiSe
  - Workload of compulsory attendance: 56

**Total module attendance time**

112 h
inf009 - Database Practical

<table>
<thead>
<tr>
<th>Module label</th>
<th>Database Practical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf009</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
</tbody>
</table>

**Applicability of the module**
- Bachelor's Programme Business Informatics (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP)
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Praktische Informatik)
- Master of Education Programme (Hauptschule and Realschule) Computing Science (Master of Education) > Mastermodule
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik

**Responsible persons**
- Grawunder, Marco (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

**Prerequisites**
- Information Systems I
- Operating system knowledge

**Skills to be acquired in this module**

The objective of this module is to gather practical experience on databases and information systems. The students get an overview of the technical realisation, implementation and optimisation of a professional database management system.

**Professional competence**
The students:
- Realise, implement and program database systems
- Program and implement database-oriented system routines
- Implement optimisation goals in the modelling phase
- Administer professional database systems (installation, maintenance and adjustment)
- Recognise database systems' performance problems and are able to fix them with according methods
- Organise and control processes of database systems

**Methodological competence**
The students:
- Propose concrete processing principles for special application classes
- Reflect on specific technologies and procedures with regard to their consequences

**Social competence**
The students:
- Solve database system problems in a team

**Self-competence**
The students:
- Acknowledge the limits of their ability to cope with pressure during the implementation and are aware of failures
- Reflect their self-perception

**Module contents**
The module “Practical Course Databases” is a related practical course of the module “Information Systems I”. The objectives of this module are special technical concepts of a database system and practical solutions in database programming and optimisation.

**Contents of this module**
- System-oriented database management programming,
- Implementation of catalogue systems,
Optimisation strategies based on parallelisation and partitioning requirements

Recommended reading

- Held Andrea (2005), Oracle 10g Hochverfügbarkeit Addison-Wesley -
- Held Andrea (2015), Oracle 12c New Features Addison Wesley

Links

- http://www-is.informatik.uni-oldenburg.de/227/

Language of instruction

German

Duration (semesters)

1 Semester

Module frequency

every winter term

Module capacity

unlimited

Module level

Type of module

Teaching/Learning method

P

Previous knowledge

Information Systems I

Operating system knowledge

Examination

Examination times

Type of examination

Final exam of module

At the end of the lecture period

Oral exam

Type of course

Exercises

SWS

4

Frequency

WiSe

On-site workload

56 h
### inf014 - Operating Systems Practical

<table>
<thead>
<tr>
<th>Module label</th>
<th>Operating Systems Practical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf014</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
<tr>
<td>Applicability of the module</td>
<td></td>
</tr>
<tr>
<td>Bachelor's Programme Business Informatics (Bachelor) &gt; Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik</td>
<td></td>
</tr>
<tr>
<td>Bachelor's Programme Computing Science (Bachelor) &gt; Akzentsetzungsbereich - Wahlbereich Informatik</td>
<td></td>
</tr>
<tr>
<td>Dual-Subject Bachelor's Programme Computing Science (Bachelor) &gt; Praktische Vertiefung (60 KP)</td>
<td></td>
</tr>
<tr>
<td>Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) &gt; Praktische Vertiefung der Informatik</td>
<td></td>
</tr>
<tr>
<td>Responsible persons</td>
<td></td>
</tr>
<tr>
<td>Theel, Oliver (module responsibility)</td>
<td></td>
</tr>
<tr>
<td>Lehrenden, Die im Modul (authorised to take exams)</td>
<td></td>
</tr>
<tr>
<td>Prerequisites</td>
<td></td>
</tr>
<tr>
<td>Information Systems I</td>
<td></td>
</tr>
<tr>
<td>Operating Systems I- Operating Systems II</td>
<td></td>
</tr>
<tr>
<td>Programming languages: C, Assembler</td>
<td></td>
</tr>
<tr>
<td>Skills to be acquired in this module</td>
<td></td>
</tr>
<tr>
<td>The aim of this module is to get practical experience in the field of analysis, design, and implementation methods of components of operating systems and their interactions.</td>
<td></td>
</tr>
</tbody>
</table>

**Professional competence**

The students:

- familiarise with complex software systems
- implement hardware-related components of operating systems
- describe parallel system operation executions
- understand the basic concepts of the programming language C++
- identify software errors systematically, especially regarding parallel software
- work in teams
- use UNIX standard software to solve problems
- recognise the advantage of working with virtual machines

**Methodological competence**

The students:

- are aware of the challenges in handling operating systems
- transfer operating system concepts to a practical context
- analyse different solutions to a problem wrt. their properties
- select the most suitable solution

**Social competence**

The students:

- solve problems in small teams
- present their solutions to all teams
- discuss their different solutions within their own team and among all teams

**Self-competence**

The students:

- accept criticism
- organise the workflows within their teams
- question their potential solutions in the light of criticism received
- identify their own shortcomings in their initial ability to successfully transfer theory to praxis

<table>
<thead>
<tr>
<th>Module contents</th>
<th>The contents of this module are:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis of a rudimentary operating system</td>
<td></td>
</tr>
<tr>
<td>Design and implementation of a process management subsystem</td>
<td></td>
</tr>
<tr>
<td>Design and implementation of process synchronisation mechanisms</td>
<td></td>
</tr>
</tbody>
</table>
- Design and implementation of a virtual memory management subsystem
- Design and implementation of a file subsystem or dialog subsystem

**Recommended reading**


**Links**

<table>
<thead>
<tr>
<th>Language of instruction</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration (semesters)</td>
<td>1 Semester</td>
</tr>
<tr>
<td>Module frequency</td>
<td>every winter term</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Reference text</td>
<td>Associated with the modules:</td>
</tr>
<tr>
<td></td>
<td>- Operating Systems I</td>
</tr>
<tr>
<td></td>
<td>- Operating Systems II</td>
</tr>
<tr>
<td></td>
<td>- Distributed Systems</td>
</tr>
</tbody>
</table>

**Module level**

**Type of module**

**Teaching/Learning method**

P

**Previous knowledge**

- Operating Systems I
- Operating Systems II
- Programming languages: C, Assembler

**Examination**

<table>
<thead>
<tr>
<th>Examination</th>
<th>Examination times</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td>At the end of the semester</td>
<td>Active participation / work report and oral exam</td>
</tr>
</tbody>
</table>

**Type of course**

<table>
<thead>
<tr>
<th>Type of course</th>
<th>Practical training</th>
</tr>
</thead>
</table>

**SWS**

4

**Frequency**

WiSe

**On-site workload**

56 h
inf018 - Media Processing

Module label: Media Processing
Module abbreviation: inf018
Credit points: 6.0 KP
Workload: 180 h

Applicability of the module

- Bachelor's Programme Biology (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Business Administration and Law (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Business Informatics (Bachelor) > Akzentsatzungsbereich Praktische Informatik und Angewandte Informatik
- Bachelor's Programme Business Informatics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Chemistry (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Computing Science (Bachelor) > Akzentsatzungsbereich - Wahlbereich Informatik
- Bachelor's Programme Computing Science (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Economics and Business Administration (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Engineering Physics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Environmental Science (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Intercultural Education and Counselling (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Mathematics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Physics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Physics, Engineering and Medicine (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Police Science (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Political Science (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Sociology (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Social Studies (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Social Work (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Elementary Mathematics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Economic Education (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Economics and Business Administration (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Elementary Mathematics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme English Studies (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
The students can explain the basics of image processing and know which algorithms exist for the basic tasks in image processing and how these are applied.

The students can apply basic methods of image processing they learned in the lecture to solve simple problems.

**Professional competence:**
- can name basic characteristics of digital media
- can explain the most common methods for encoding and compressing images, video and audio
- can describe basic procedures for image enhancement, feature extraction, feature description, image analysis and image comprehension

**Methodological competence:**
- can recognize and evaluate image properties and decide for suitable image processing methods
- can select existing software packages for simple image processing problems, as well as use and customize them for their specific task
- can implement simple image and media processing functions in a
higher programming language (e.g., C++)

Social competence
The students:

- can plan, implement, and document a software project in team work
- can present the results of their work to an audience and adequately respond to criticism and questions

Self competence
The students:

- can accept and learn from mistakes made during the process of implementation

Module contents
The lecture covers the technologies of media processing. In particular, the lecture focuses on image processing chain from digital imaging, through image pre-/and postprocessing, and image storage to image analysis. In addition to compression techniques and color space theory (RGB, HSV, YUV, CIEXYZ, ...), the topics of the lecture include image enhancement, feature extraction, feature description, image analysis and image comprehension. The lecture furthermore discusses the encoding and analysis of video and audio.

Recommended reading
- Literatur im Handapparat der Abteilung in der Bibliothek.
- Linkliste im Lernmanagementsystem zu den einzelnen Themen der Vorlesung.

Links
https://uol.de/en/media-informatics/teaching/courses

Language of instruction
German

Duration (semesters)
1 Semester

Module frequency
every winter term

Module capacity
12

Reference text

Type of module
Teaching/Learning method
1VL + 1Ü

Previous knowledge
Solid programming skills in Java and/or C++, practical informatics. Interest in media processing

Examination
The presentation of the practical project on a project day of all small groups takes place directly after the lecture period. The oral examination takes place in the first two weeks after the end of the lecture period. Any post-examinations will take place at the end of the lecture period. The exact schedule can be found on the department's web pages as well as the information in the learning management system Stud.IP.

Type of examination
Project and oral Exam or project and written exam

Type of course
Comment
SWS
Frequency
Workload of compulsory attendance

Lecture
2
WiSe
28

Exercises
2
WiSe
28

Total module attendance time
56 h
inf021 - Advanced Java Technologies

Module label: Advanced Java Technologies
Module abbreviation: inf021
Credit points: 6.0 KP
Workload: 180 h

Applicability of the module:
- Bachelor's Programme Business Informatics (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP)
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik

Responsible persons:
- Boles, Dietrich (module responsibility)

Prerequisites:
- useful knowledge: Object-oriented programming

Skills to be acquired in this module:
The objective of this module is to introduce advanced concepts and technologies of the Java Standard Edition. The students will be able to use the technologies to implement large applications.

Professional competence:
The students:
- name the essential packages of the JDK class library
- structure large programs properly and implement them extensively
- set up own Java class libraries
- look up required classes in the JDK-Library and solve problems with these classes
- structure their programs properly
- understand and interpret large programs of other students
- evaluate the quality of large programs related to their maintainability, reusability and expandability

Methodological competence:
The students:
- search for solutions to specific problems in the internet independently

Social competence:
The students:
- discuss own and solutions of other students

Self-competence:
The students:
- reflect their problem-solving behaviour and take up new solutions, e.g. from the internet

Module contents:
A selection of the following subjects is presented during the lectures:
- GUI (AWT, Swing, JavaFX)
- Java-Basics and Collection-API
- Graphics and multimedia
- Events
- Model-View-Control (MVC)
- Threads
- Internationalization, localization
- Reflection
- IO, Files
- Tools (compiler, classloader, printer, ...)
- Storage technologies (XML and serialization)
- Distributed programming (sockets and RMI)
- Databases (JDBC)
- Compression
- Security concepts
Alternatively, a single topic is explored in depth.
As part of the exercises, individual programming tasks or a larger programming task will be worked on. The tasks are related to the topic of the individual lecture contents.

<table>
<thead>
<tr>
<th>Recommended reading</th>
<th>list of links in the learning management system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Links</td>
<td></td>
</tr>
<tr>
<td>Language of instruction</td>
<td>German</td>
</tr>
<tr>
<td>Duration (semesters)</td>
<td>1 Semester</td>
</tr>
<tr>
<td>Module frequency</td>
<td>every semester</td>
</tr>
<tr>
<td>Module capacity</td>
<td>12</td>
</tr>
<tr>
<td>Reference text</td>
<td></td>
</tr>
<tr>
<td>Module level</td>
<td></td>
</tr>
<tr>
<td>Type of module</td>
<td></td>
</tr>
<tr>
<td>Teaching/Learning method</td>
<td>1V + 1Ü</td>
</tr>
<tr>
<td>Previous knowledge</td>
<td>Object-oriented programming</td>
</tr>
<tr>
<td>Examination</td>
<td>Examination times</td>
</tr>
<tr>
<td>Final exam of module</td>
<td>throughout the semester</td>
</tr>
<tr>
<td>Type of examination</td>
<td>practical exercises</td>
</tr>
</tbody>
</table>

**Type of course**

<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>2</td>
<td>SoSe oder WiSe</td>
<td>56</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>2</td>
<td>SoSe oder WiSe</td>
<td>28</td>
</tr>
<tr>
<td><strong>Total module attendance time</strong></td>
<td></td>
<td></td>
<td></td>
<td>84 h</td>
</tr>
<tr>
<td>Module label</td>
<td>Computer Engineering Practical</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Module abbreviation</td>
<td>inf202</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Applicability of the module**

- Bachelor's Programme Biology (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Business Administration and Law (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Business Informatics (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Chemistry (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Comparative and European Law (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Computing Science (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Economics and Business Administration (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Education (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Environmental Science (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Intercultural Education and Counselling (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Mathematics (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Physics (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Physics, Engineering and Medicine (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Social Studies (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Sustainability Economics (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Art and Media (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Biology (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Chemistry (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Dutch Linguistics and Literary Studies (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Economic Education (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Elementary Mathematics (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme English Studies (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Gender Studies (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme General Education (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme German Studies (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme History (Bachelor) > Fachnahe Angebote Informatik
- Dual-subject bachelor's programme Low German (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Material Culture: Textiles (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Mathematics (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Music (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Philosophy / Values and Norms (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Physics (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Politics-Economics (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Protestant Theology and Religious Education (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Slavic Studies (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Social Studies (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Sport Science (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Special Needs Education (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Technology (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft (Bachelor) > Fachnahe Angebote Informatik
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik)
- Master of Education Programme (Hauptschule and Realschule) Computing Science (Master of Education) > Mastermodule
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik

**Responsible persons**
- Fränzle, Martin Georg (module responsibility)
- Janßen, Detlef (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

**Prerequisites**
Recommendation: inf200 "Fundamentals of Computer Engineering"

**Skills to be acquired in this module**
This course enables students to analyze information technology systems, understand individual components of computers, design and optimize them, and discuss domain-specific hardware design in a qualified manner.

**Professional competences**
The students
- describe individual components of computers
- design and optimize individual components of computers
- design and optimize automata specify and imply autonomous systems

**Methodological competence**
The students
- synthesize computer architectures
- can transfer methods of hardware design to different systems

**Social competence**
The students
- discuss hardware in a qualified manner

**Self-competence**
The students
- are able to clearly distinguish their level of knowledge from professionals of related disciplines

**Module contents**
This module is the practical part of the course Introduction to Computer Engineering.

**Recommended reading**
- Script for the course
- Patterson, D.A., Hennessy, J.L.: Computer Organisation and Design: The Hardware/Software Interface

**Links**

**Language of instruction**
German
<table>
<thead>
<tr>
<th><strong>Duration (semesters)</strong></th>
<th>1 Semester</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Module frequency</strong></td>
<td>every summer term</td>
</tr>
<tr>
<td><strong>Module capacity</strong></td>
<td>unlimited</td>
</tr>
<tr>
<td><strong>Module level</strong></td>
<td>Type of module</td>
</tr>
<tr>
<td><strong>Teaching/Learning method</strong></td>
<td>1 P</td>
</tr>
<tr>
<td><strong>Previous knowledge</strong></td>
<td>Recommendation: inf200 Fundamentals of Computer Engineering</td>
</tr>
<tr>
<td><strong>Examination</strong></td>
<td>Examination times</td>
</tr>
<tr>
<td><strong>Final exam of module</strong></td>
<td>At the end of the lecture term</td>
</tr>
<tr>
<td><strong>Type of course</strong></td>
<td>Practical training</td>
</tr>
<tr>
<td><strong>SWS</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Frequency</strong></td>
<td>SoSe</td>
</tr>
<tr>
<td><strong>On-site workload</strong></td>
<td>56 h</td>
</tr>
</tbody>
</table>
inf800 - Proseminar in Computer Science

<table>
<thead>
<tr>
<th>Module label</th>
<th>Proseminar in Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf800</td>
</tr>
<tr>
<td>Credit points</td>
<td>3.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>90 h</td>
</tr>
</tbody>
</table>

**Applicability of the module**
- Bachelor's Programme Business Informatics (Bachelor) > Aufbaucurriculum - Pflichtbereich
- Bachelor’s Programme Computing Science (Bachelor) > Aufbaumodule
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP)
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik

**Responsible persons**
- Diethelm, Ira (module responsibility)
- Nieße, Astrid (module responsibility)
- Sauer, Jürgen (module responsibility)
- Lehrenden, Die im Modul (Module counselling)

**Prerequisites**
- Studierende im den Bachelor-Studiengängen der Informatik sowie Master of Education Informatik

**Skills to be acquired in this module**
Supported by a lecturer the students familiarise with a given topic by literature research. They understand and evaluate the relevance of the literature. After this evaluation the students present and discuss their solutions academically.

**Professional competence**
The students:
- characterise and apply computer science basics (algorithms, data structures, programming, basics of practical, technical and theoretical computer science)
- define and describe essential mathematical, logical and physical basics of computer science
- define and illustrate the core disciplines of computer science (theoretical, practical and technical computer science)

**Methodological competence**
The students:
- examine problems, use formal methods to phrase them and analyze them appropriately
- evaluate problems by the use of technical and scientific literature
- reflect on a scientific topic and write a scientific seminar paper under guidance and present their findings

**Social competence**
The students:
- communicate considerately and appropriately with users and experts
- use presentation methods

**Self-competence**
The students:
- plan their informatical actions independently
- reflect their contributions critically and discuss them with users and experts
- collect and update their knowledge independently

**Module contents**
- according to the assigned task

**Recommended reading**

**Links**

**Language of instruction**
- German

**Duration (semesters)**
- 1 Semester

**Module frequency**
- semi-annual

**Module capacity**
- unlimited

**Reference text**
- Choose one of the seminaire courses of the module.
<table>
<thead>
<tr>
<th>Module level</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of module</td>
<td></td>
</tr>
<tr>
<td>Teaching/Learning method</td>
<td>1S</td>
</tr>
<tr>
<td>Previous knowledge</td>
<td>none</td>
</tr>
<tr>
<td>Examination</td>
<td>Examination times</td>
</tr>
<tr>
<td>Final exam of module</td>
<td>Type of examination</td>
</tr>
<tr>
<td>Am Ende des Semesters und nach Absprache</td>
<td>Presentation</td>
</tr>
<tr>
<td>Type of course</td>
<td>Seminar</td>
</tr>
<tr>
<td>SWS</td>
<td>2</td>
</tr>
<tr>
<td>Frequency</td>
<td>SoSe oder WiSe</td>
</tr>
<tr>
<td>On-site workload</td>
<td>28 h</td>
</tr>
</tbody>
</table>
### inf803 - Special Topics in Computer Science I

<table>
<thead>
<tr>
<th>Module label</th>
<th>Special Topics in Computer Science I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf803</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
</tbody>
</table>
| Applicability of the module | - Bachelor's Programme Business Informatics (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik  
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik  
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP)  
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik |
| Responsible persons | Lehrende, Die im Modul (authorised to take exams) |
| Prerequisites | No participant requirements |
| Skills to be acquired in this module | This module integrates current computer science developments within appropriate study courses. **Professional competence**  
The students:  
- know recent technological or scientific computer science developments  
- transfer computer science methods and development models to IT application area requirements  
- evaluate the possibilities and limitations of computer science methods and tools and apply them appropriately  
**Methodological competence**  
The students:  
- review problems, formulate them with formal models and explore them appropriately  
- identify and present (one or more) computer science problem solutions  
- select and evaluate appropriate tools and methods  
- examine problems with technical and scientific literature  
**Social competence**  
The students:  
- work in a team  
**Self-competence**  
The students:  
- plan their informatical actions independently |
<p>| Module contents | According to the assigned task |
| Recommended reading | Werden in der zugeordneten Lehrveranstaltung bekannt gegeben. |
| Links | |
| Languages of Instruction | German, English |
| Duration (semesters) | 1 Semester |
| Module frequency | semi-annual |
| Module capacity | unlimited |
| Module level | |
| Type of module | |
| Teaching/Learning method | 2 events from V, Ü, S, P, PR |
| Previous knowledge | none |
| Examination | Examination times | Type of examination |
| Final exam of module | Exercises or presentation or oral exam or written exam |
| Type of course | VA-Auswahl |</p>
<table>
<thead>
<tr>
<th>SWS</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>SoSe oder WiSe</td>
</tr>
<tr>
<td>On-site workload</td>
<td>56 h</td>
</tr>
</tbody>
</table>
inf804 - Special Topics in Computer Science II

Module label | Special Topics in Computer Science II
--- | ---
Module abbreviation | inf804
Credit points | 6.0 KP
Workload | 180 h

Applicability of the module
- Bachelor's Programme Business Informatics (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP)
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik

Responsible persons
- Lehrenden, Die im Modul (authorised to take exams)

Prerequisites
- No participant requirement

Skills to be acquired in this module
This module integrates current computer science developments within appropriate study courses. **Professional competence**
The students:
- know recent technological or scientific computer science developments
- transfer computer science methods and development models to IT application area requirements
- evaluate the possibilities and limitations of computer science methods and tools and apply them appropriately

**Methodological competence**
The students:
- review problems, formulate them with formal models and explore them appropriately
- identify and present (one or more) computer science problem solutions
- select and evaluate appropriate tools and methods
- examine problems with technical and scientific literature

**Social competence**
The students:
- work in a team

**Self-competence**
The students:
- plan their informatical actions independently

Module contents
According to the assigned task

Recommended reading
Werden in der zugeordneten Lehrveranstaltung bekannt gegeben

Links

Languages of instruction
German, English

Duration (semesters)
1 Semester

Module frequency
semi-annual

Module capacity
unlimited

Module level

Type of module

Teaching/Learning method
2 events from V, Ü, S, P, PR

Previous knowledge
none

Examination

Examination times

Type of examination
Exercises or presentation or oral exam or written exam

Final exam of module

Type of course
VA-Auswahl
<table>
<thead>
<tr>
<th><strong>SWS</strong></th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Frequency</strong></td>
<td>SoSe oder WiSe</td>
</tr>
<tr>
<td><strong>On-site workload</strong></td>
<td>56 h</td>
</tr>
</tbody>
</table>
**inf808 - Current Topics in Computer Science**

<table>
<thead>
<tr>
<th>Module label</th>
<th>Current Topics in Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf808</td>
</tr>
<tr>
<td>Credit points</td>
<td>3.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>90 h</td>
</tr>
</tbody>
</table>

**Applicability of the module**
- Bachelor's Programme Business Informatics (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP)
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik

**Responsible persons**
- Nieße, Astrid (module responsibility)
- Sauer, Jürgen (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

**Prerequisites**
- Keine

**Skills to be acquired in this module**
This module integrates current computer science developments within appropriate study courses.

**Professional competence**
The students:
- Know recent technological or scientific computer science developments
- Transfer computer science methods and development models to IT application area requirements
- Evaluate the possibilities and limits of computer science methods and tools and apply them appropriately

**Methodological competence**
The students:
- Review problems, formulate them with formal models and explore them appropriately
- Identify and present (one or more) computer science problem solutions
- Select and evaluate appropriate tools and methods
- Reflect on a scientific topic and write a scientific seminar paper under guidance and present their findings

**Social competence**
The students:
- Use presentation methods purposefully

**Self-competence**
The students:
- Plan their informatical actions independently
- Reflect their contributions critically and discuss them with users and experts
- Collect and update their knowledge independently

**Module contents**
According to the assigned task

**Recommended reading**
According to the assigned task

**Language of instruction**
German

**Duration (semesters)**
1 Semester

**Module frequency**
unregelmäßig

**Module capacity**
unlimited

**Module level**

**Teaching/Learning method**

**Previous knowledge**
<table>
<thead>
<tr>
<th>Examination</th>
<th>Examination times</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td></td>
<td>Exercises or presentation or oral or written exam</td>
</tr>
<tr>
<td>Type of course</td>
<td>VA-Auswahl</td>
<td></td>
</tr>
<tr>
<td>SWS</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>SoSe oder WiSe</td>
<td></td>
</tr>
<tr>
<td>On-site workload</td>
<td>28 h</td>
<td></td>
</tr>
</tbody>
</table>
Wahlpflicht Technische Informatik (30 KP)

inf200 - Computer Engineering I

<table>
<thead>
<tr>
<th>Module label</th>
<th>Computer Engineering I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf200</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
<tr>
<td>Applicability of the module</td>
<td></td>
</tr>
<tr>
<td>- Bachelor’s Programme Computing Science (Bachelor) &gt; Basismodule</td>
<td></td>
</tr>
<tr>
<td>- Bachelor’s Programme Mathematics (Bachelor) &gt; Nebenfachmodule</td>
<td></td>
</tr>
<tr>
<td>- Dual-Subject Bachelor's Programme Computing Science (Bachelor) &gt; Basismodule</td>
<td></td>
</tr>
<tr>
<td>- Dual-Subject Bachelor's Programme Computing Science (Bachelor) &gt; Wahlpflicht Technische Informatik (30 KP)</td>
<td></td>
</tr>
<tr>
<td>- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) &gt; Akzentsetzungsbereich</td>
<td></td>
</tr>
<tr>
<td>Responsible persons</td>
<td>Rauh, Andreas (module responsibility)</td>
</tr>
<tr>
<td>Lehrenden, Die im Modul (authorised to take exams)</td>
<td></td>
</tr>
<tr>
<td>Prerequisites</td>
<td>No participant requirements</td>
</tr>
<tr>
<td>Skills to be acquired in this module</td>
<td></td>
</tr>
</tbody>
</table>

The students learn to understand the construction of digital circuits and digital computers. They know the technological parameters, the state of the art technologies, and the developments characterizing current and future design paradigms for digital hardware. They learn to understand the concepts underlying current computer architectures and are able to explain how such architectures execute programs. Successful participants will be able to analyze computer architectures as a whole, to understand in depth, to analyze, and to optimize their hardware components, and to discuss the properties induced by selecting design alternatives.

Professional competences

The Students:

- identify fundamental concepts of the construction of digital computer systems, the internal number representation, and analysis of combinational logic as well as their optimization.

Methological competences

The Students:

- analyze computer architectures on the basis of their individual components
- design and optimize digital hardware components
- transfer systematic approaches of hardware design to unknown design problems

Social competences

The Students:

- present their understanding of the functional principles of digital computer systems

Self-competences

The Students:

- critically reflect on the results of exercises and recognize limitations of different approaches to the design of digital computer systems

Module contents

This module is the first part of the introduction to computer engineering. It explains the construction principles of computers, from the implementation of an easy Instruction Set Architecture, over fundamental techniques for coding and representation of numbers, program execution on machine level, basics of logics and analysis of functions of combinational logic as well as their optimization.

Recommended reading

- Lecture Notes

<table>
<thead>
<tr>
<th>Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language of instruction</td>
</tr>
<tr>
<td>Duration (semesters)</td>
</tr>
<tr>
<td>Module frequency</td>
</tr>
<tr>
<td>Module capacity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of module</td>
</tr>
<tr>
<td>Teaching/Learning method</td>
</tr>
<tr>
<td>Previous knowledge</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examination times</td>
</tr>
<tr>
<td>Final exam of module</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3</td>
<td>WiSe</td>
<td>42</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1</td>
<td>WiSe</td>
<td>14</td>
</tr>
</tbody>
</table>

| Total module attendance time | 56 h |
inf201 - Computer Engineering II

Module label: Computer Engineering II
Module abbreviation: inf201
Credit points: 6.0 KP
Workload: 180 h

Applicability of the module
- Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik)
- Master of Education Programme (Hauptschule and Realschule) Computing Science (Master of Education) > Mastermodule
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Akzentsetzungsbereich

Responsible persons
- Rauh, Andreas (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

Prerequisites
Knowledge of the module "Computer Engineering I"

Skills to be acquired in this module

The module qualifies students to analyze computer architectures, understand computer components, design and optimize computers and components, and to discuss domain-specific hardware design.

Professional Competences
The students:
- describe computer components
- design and optimise computer components
- describe and analyse electric circuits

Methodological Competences
The students:
- analyse computer architectures
- get familiar with fundamentals of the analysis and synthesis of flipflops and automata
- get familiar with foundations of the analysis of electrical circuits

Social Competences
The students:
- discuss computer hardware and manufacturing processes competently
- are able to transfer their knowledge of hardware design to other domains different from computer science

Self Competences
The students:
- critically reflect the results of exercises and acknowledge limitations of various approaches for the design of computer systems

Module contents
This module is the second part of the introduction to computer engineering. It explains sequential circuits (e.g. flipflops and automata), arithmetic and logical computer components, registers and memory, basics of computer communication as well as electrotechnical foundations.

Recommended reading
- Lecture Notes
- Oberschelp, W., Vossen, G.: Rechneraufbau und Rechnerstrukturen; Oldenbourg Verlag

Links
<table>
<thead>
<tr>
<th>Language of instruction</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration (semesters)</td>
<td>1 Semester</td>
</tr>
<tr>
<td>Module frequency</td>
<td>annual</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Module level</td>
<td></td>
</tr>
<tr>
<td>Type of module</td>
<td></td>
</tr>
<tr>
<td>Teaching/Learning method</td>
<td>1VL + 1Ü</td>
</tr>
<tr>
<td>Previous knowledge</td>
<td>Knowledge of the module &quot;Computer Engineering I&quot;</td>
</tr>
<tr>
<td>Examination</td>
<td>Examination times</td>
</tr>
<tr>
<td>Final exam of module</td>
<td>At the end of the lecture period</td>
</tr>
<tr>
<td>Type of examination</td>
<td>Written or oral Exam</td>
</tr>
<tr>
<td>Type of course</td>
<td>Comment</td>
</tr>
<tr>
<td>Lecture</td>
<td>SWS</td>
</tr>
<tr>
<td></td>
<td>Frequency</td>
</tr>
<tr>
<td></td>
<td>Workload of compulsory</td>
</tr>
<tr>
<td></td>
<td>attendance</td>
</tr>
<tr>
<td>Lecture</td>
<td>3 SoSe</td>
</tr>
<tr>
<td>Exercises</td>
<td>1 SoSe</td>
</tr>
<tr>
<td>Total module attendance time</td>
<td>56 h</td>
</tr>
</tbody>
</table>
inf203 - Embedded Systems I

Module label
Embedded Systems I

Module abbreviation
inf203

Credit points
6.0 KP

Workload
180 h

Applicability of the module
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
- Master's Programme Engineering of Socio-Technical Systems (Master) > Embedded Brain Computer Interaction
- Master's Programme Engineering of Socio-Technical Systems (Master) > Human-Computer Interaction
- Master's Programme Engineering of Socio-Technical Systems (Master) > Systems Engineering

Responsible persons
- Fränzle, Martin Georg (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

Prerequisites
- Basics of technical computer science
- Computer Engineering

Skills to be acquired in this module

Professional competence
The students:
- name functional and non-functional requirements to specify embedded systems
- discuss design space and associated embedded systems design methods
- name control and feedback control systems' core concepts
- characterise the fundamental digital signal processing algorithms

Methodological competence
The students:
- design and develop embedded feedback control systems with modelling tools
- implement an embedded hardware/software system according to a given specification
- analyze various specification languages according to different properties

Social competence
The students:
- implement solutions to given problems in teams
- present results of computer science problems to groups
- organize themselves as a team to solve a larger problem using project management methods

Self-competence
The students:
- acknowledge the limits of their ability to cope with pressure during the implementation process of systems
- solve exercises self-responsibly

Module contents
Embedded systems support complex feedback problems, control problems and data processing tasks. They have an important value creation potential for telecommunications, production management, transport and electronics. The functionality of embedded systems is realised by the integration of processors, special hardware and software. The embedded systems design is influenced by the heterogeneity of system architectures, the complexity of systems and technical and economic requirements. This module gives an overview of embedded systems and their design. The process of digital signals is
especially important for telecommunications and multimedia. For this purpose, the module introduces digital signal processing algorithms. The principles of feedback control are introduced by exemplary transport applications. Subsequently, the module provides the specifications and language characteristics of the embedded system design. For this purpose, graphical data-flow modelling languages (for instance Simulink) and control-flow specifications (for instance State Charts) are presented. The module closes with the concepts of possible architectures and communication models. Hands-on exercises with the tools Matlab/Simulink/StateFlow support the module contents.

**Recommended reading**


**Secondary literature:**

- Artikelserie zum MPEG-2-Standard 3/94 10/94 und das Tutorial "Digitale Bildcodierung" 1/92 1/93, beides in "Fernseh- und Kinotechnik" (BIS: Z elt ZA 1536)

**Links**

<table>
<thead>
<tr>
<th>Language of instruction</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration (semesters)</td>
<td>1 Semester</td>
</tr>
<tr>
<td>Module frequency</td>
<td>annual</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
</tbody>
</table>

**Reference text**

This module is compulsory for students who are specialising in "Eingebettete Systeme und Mikrorobotik". **Associate with the modules:** In the module "Eingebettete Systeme II" additional relevant topics such as design processes, HW/SW-Partitioning, High-Level-Synthesis and Hardware description languages are discussed. The modules Eingebettete Systeme I und II offer cross-references to the module "Rechnerarchitektur", "Realzeitbetriebssysteme" and semantic orientated modules of theoretical computer science. It is possible to enhance the knowledge of embedded systems design by attending the modules "System Level Design" and "Low energy System Design".

**Module level**

<table>
<thead>
<tr>
<th>Type of module</th>
<th>1VL + 1Ü</th>
</tr>
</thead>
</table>
| Previous knowledge | - Basics of technical computer science  
|                            - Computer Engineering |

**Examination**

<table>
<thead>
<tr>
<th>Examination times</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Written or oral exam</td>
<td>50 / 74</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Final exam of module</th>
<th>At the end of the semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of course</td>
<td>Comment</td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
</tr>
<tr>
<td>Lecture</td>
<td></td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
</tr>
<tr>
<td><strong>Total module attendance time</strong></td>
<td></td>
</tr>
</tbody>
</table>
inf204 - Embedded Systems II

Module label | Embedded Systems II
---|---
Module abbreviation | inf204
Credit points | 6.0 KP
Workload | 180 h

Applicability of the module
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik)
- Master's Programme Engineering of Socio-Technical Systems (Master) > Embedded Brain Computer Interaction
- Master's Programme Engineering of Socio-Technical Systems (Master) > Systems Engineering

Responsible persons
- Fränzle, Martin Georg (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

Prerequisites
- No participant requirement

Skills to be acquired in this module
- The module provides an introduction to digital embedded systems design.

**Professional competence**
The students:
- name embedded systems architectures
- name specific hardware components and -architecture designs, particularly processor designs
- characterise the design spaces and associated embedded systems design techniques
- decompose subcomponents of feedback control systems and implement their tasks in different design spaces
- develop software-/hardware components
- describe fault-tolerance architecture principles
- describe real-time and safety requirements analysing techniques
- characterise hardware synthesis

**Methodological competence**
The students:
- estimate the consequences of design decisions in terms of energy usage, performance and reliability component allocations, and designs
- implement an embedded hardware/software system according to a given specification
- model hardware with a hardware description languages - analyze Hardware/Software systems using event-bases simulation

**Social competence**
The students:
- implement solutions to given problems in teams
- present results of computer science problems to groups
- organize themselves as a team to solve a larger problem using project management methods

**Self-competence**
The students:
- acknowledge the limits of their ability to cope with pressure during the implementation process of systems
- deal self responsibly with exercises

**Module contents**
Embedded systems support complex feedback problems, control problems and data processing tasks. They have an important value creation potential for telecommunications, production management, transport and electronics. The functionality of embedded systems is realised by the integration of processors, special hardware and software. The embedded systems design is influenced by the heterogeneity of system architectures, the complexity of systems and technical and economic requirements. This module is the continuation of the module “Eingebettete Systeme I” and deals with different architectures of embedded systems and processors. The module provides system partitioning methods and the synthesis of hardware components. Hands-on exercises with development tools, hardware description languages and simulation support the
module contents.

**Recommended reading**


**Secondary literature:**


**Links**

- Language of instruction: English
- Duration (semesters): 1 Semester
- Module frequency: annual
- Module capacity: unlimited
- Reference text: This module is supposed to be a compulsory module for students who are specialising in "Eingebettete Systeme und Mikrorobotik".

**Module level**

- Type of module: Teaching/Learning method 1VL + 1Ü
- Previous knowledge: none

**Examination**

- Examination times: At the end of the lecture times
- Type of examination: Written or oral Exam

**Final exam of module**

<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3</td>
<td>SoSe</td>
<td>42</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1</td>
<td>SoSe</td>
<td>14</td>
</tr>
</tbody>
</table>

**Total module attendance time**: 56 h
inf205 - Formal Methods in Embedded System Design

<table>
<thead>
<tr>
<th>Module label</th>
<th>Formal Methods in Embedded System Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf205</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
</tbody>
</table>

**Applicability of the module**

- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik)

**Responsible persons**

- Fränzle, Martin Georg (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

**Prerequisites**

Sound basic knowledge in mathematical logic, discrete mathematics, automata and computability theory as taught in the modules "Discrete Structures" and "Theoretical Computer Science I + II". In addition, programming skills as acquired in the "Programming Course".

Justification:
The methods presented in the lecture are based on an operationalization of semantics by reduction to logical encodings and mechanized testing of logical statements. An understanding of these contents as well as their tool-technical implementation requires the basics from the aforementioned courses.

**Skills to be acquired in this module**

The module provides an overview over semantic models for reactive systems, real-time systems and hybrid systems, as well as examples of corresponding specification logics. It explains state-exploratory verification procedures in both explicit and symbolic variants. The knowledge acquired can be employed in all domains requiring the development of reliable software and hardware systems is concerned

**Professional competences**

The students:

- make a sound judgement on the scope of the certificates that can be obtained with formal methods
- assess the suitability of available verification tools for a particular problem and system class
- use automatic analysis tools on real systems, interpret the results obtained and subsequently improve the system under investigation in an informed and targeted manner
- prepare system models for automatic analysis procedures and encode them symbolically (or otherwise)
- design and implement their own verification algorithms

**Methodological competences**

The students:

- master the mathematical modelling of complex and heterogeneous dynamical systems
- know relevant mathematical models of dynamic systems and can instantiate them to new problem classes

**Social competences**

The students:

- together in a team develop and implement basic algorithms of automatic verification
- discuss the advantages and disadvantages of algorithmic alternatives and different formalisations

**Self competences**

The Students:

- can assess their technical and methodological understanding
- reflect on their problem-solving competence with reference to the procedures and methods presented

**Module contents**

Embedded computer systems maintain constant interaction with their
environment. This induces interaction sequences that are difficult to predict, which complicates the construction and validation of such systems. Akin to the use of structural models for rigorous validation of building layouts in the construction industry, formal models in computer science have consequently been developed for analysis of various aspects of computing systems in general and embedded systems in particular. They cover execution time, energy demand, or possible system dynamics of embedded systems. They represent the respective aspect of the system in a closed form and thus allow for the - often fully automatic - derivation of reliable certificates that apply to any interaction scenario with the environment. This is in contrast to methods of testing or profiling, which only test selected scenarios and thus can only provide limited coverage.

In this module, various such models are explained and methods for their fully automatic analysis - i.e., derivation certificates - or synthesis - i.e., automatic generation of correct system designs - from such models are explained and demonstrated in their application.

The exercises provide opportunity for deepening understanding through hands-on experience with domain-specific modelling and verification tools, as well as by creating a (small) fully automated verification tool yourself in a guided process.

In the lectures, the semantic, logical, and algorithmic basics of automatic analysis of embedded software systems are taught. The primary form of instruction is the media-supported lecture as well as the didactic question-answer game.

In the exercise classes, the knowledge acquired in the lecture is deepened and put into practice. For this purpose, in the first half of the semester, exercises are set fortnightly, the completion of which in small groups encourages independent testing of the individual understanding of the topic and peer teaching. In the second half of the semester, a larger tool development task is set, also to be pursued on in small groups of 3 students each. The work on these projects spans the entire second half of the semester and offers the possibility of project-oriented learning. In this phase, the exercise classes serve as consultation time with the lecturers; in particular, solution approaches and problems can be presented and discussed.

Recommended reading


Links

Languages of instruction: German, English

Duration (semesters): 1 Semester

Module frequency: annual

Module capacity: unlimited

Module level

Type of module: Teaching/Learning method: 1VL + 1Ü + 1P

Previous knowledge: Sound basic knowledge in mathematical logic, discrete mathematics, automata and computability theory as taught in the modules "Discrete Structures" and "Theoretical Computer Science I + II". In addition, programming skills as acquired in the "Programming Course".

Justification: The methods presented in the lecture are based on an operationalization of semantics by reduction to logical encodings and mechanized testing of logical statements. An understanding of these contents as well as their tool-technical implementation requires the basics from the aforementioned courses.

Examination

Examination times

Type of examination

Final exam of module

- 1st deadline: Submission of the semester project incl. one week after the end of the lecture period of the lecture period; followed
Examination times

by a colloquium and a final discussion

- 2nd deadline: Repeat of the submission of the semester project incl. written elaboration two weeks before the beginning of the following semester followed by colloquium and final discussion

<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3</td>
<td>WiSe</td>
<td>42</td>
</tr>
<tr>
<td>Übung und Projekt</td>
<td></td>
<td>1</td>
<td>WiSe</td>
<td>14</td>
</tr>
</tbody>
</table>

**Total module attendance time**

56 h
inf207 - Electrical Engineering

<table>
<thead>
<tr>
<th>Module label</th>
<th>Electrical Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf207</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
</tbody>
</table>

**Applicability of the module**
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik)
- Master of Education Programme (Hauptschule and Realschule) Computing Science (Master of Education) > Mastermodule
- Master's Programme Computing Science (Master) > Interdisziplinäre Module

**Responsible persons**
- Hein, Andreas (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

**Prerequisites**
- Module Analysis II or Numerics

**Skills to be acquired in this module**

**Professional competence:**
The students:
- analyse linear electrical networks (direct current and alternating current)
- name basic concepts to calculate and to use electrical and magnetic fields
- list the characteristics of simple electrical elements (two terminal networks)
- calculate the parameters of simple electrical networks/wirings
- apply computer based analysing tools
- design and implement simple networks/wirings

**Methodological competence:**
The students:
- transfer calculation methods onto complex dynamic systems
- implement electrical system models

**Social competence:**
The students:
- present solutions for specific questions

**Self-competence:**
The students:
- reflect their solutions by using methods learned in this course

**Module contents**
- Basic concepts (electric dimensions and units)
- Network elements
- Calculation of linear direct current networks (Ohms law, Kirchhoff's circuit law, superposition principle)
- Characteristics, calculations and representations of electric and magnetic fields
- Construction elements (capacitor and coil)
- Extensions of periodical dimensions dependent on time, pointer representation, calculations with complex root-mean-square value pointers

**Recommended reading**

**essential:**
- slides

**recommended:**
- Hagmann, G.: Aufgabensammlung zu den Grundlagen der
<table>
<thead>
<tr>
<th>Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language of instruction</td>
</tr>
<tr>
<td>Duration (semesters)</td>
</tr>
<tr>
<td>Module frequency</td>
</tr>
<tr>
<td>Module capacity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of module</td>
</tr>
<tr>
<td>Teaching/Learning method</td>
</tr>
<tr>
<td>Previous knowledge</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examination</th>
<th>Examination times</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td>At the End of the Semester</td>
<td>Hands-on exercises / written exam or oral exam</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3</td>
<td>SoSe</td>
<td>42</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1</td>
<td>SoSe</td>
<td>14</td>
</tr>
</tbody>
</table>

| Total module attendance time | 56 h |
inf208 - Microrobotics and Microsystems Technology

Module label: Microrobotics and Microsystems Technology
Module abbreviation: inf208
Credit points: 6.0 KP
Workload: 180 h

Applicability of the module
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik)
- Master's Programme Computing Science (Master) > Interdisziplinäre Module

Responsible persons
- Fatikow, Sergej (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

Prerequisites
No participant requirements

Skills to be acquired in this module
Within the last few years, microrobotics and microsystem technology (MST) have become a focus of interest to industry and evolved into an important field with great application potential. It plays a decisive role for industry to be competitive in many areas such as medicine, production engineering, biotechnology, environmental technology, automotive products, etc. Despite of the growing interest in this new technology, there is hardly any book or lecture course that treats microrobotics and MST in a coherent and comprehensive way. This course is an attempt of the Microrobotics and Control Engineering Division (AMiR) to give students a systematic introduction to microrobotics and MST. It discusses all important aspects of this rapidly expanding technology, its diversity of products and fields of application. The course contains an overview of numerous ideas of new devices and the problems of manufacturing them.

Professional competence:
The students:
- name the ideas, challenges and activities of microrobotics and microsystem technology
- describe the microrobotics and MST applications
- characterise MST methods
- name microsensor functionality
- characterise microsensor examples
- discuss MST terms of information technology
- classify microrobotics

Methodological competence
The students:
- discover interdisciplinary connections and links between scientific and technical fields of research and development
- learn technical abstraction of complex contexts

Social competence
The students:
- solving problems partially as group
- present their solutions and approaches to the group

Self-competence
The students:
- reflect their knowledge of technical computer science
- learn to expand on their professional competence independently

Module contents
Ideas and problems of microrobotics and MST:
- applications:
- techniques of MST:
- silicon-based micromechanics:
- LIGA technology:

Microactuators:
- principles and examples (electrostatic, piezoelectric, magnetostrictive, electromagnetic, SMA-based, thermomechanical, electromechanical
and other actuators);

**Microsensors:**
- principles and examples (force and pressure, position and speed, acceleration, biological and chemical, temperature and other sensors);
- MST and information processing;
- microsystem design and simulation;
- classification of microrobots;
- coarse positioning of a microrobot;
- fine positioning of a microrobot;

**Handling of microparts:**
- problems and solutions;
- micro grasp techniques;
- microassembly;

**Process automation by microrobots:**
- desktop robot cell in SEM

---

**Recommended reading**

**Essential:**
- Vorlesungsskript in Buchform

**Recommended:**

**Secondary Literature:**
- Eibel, Th.: Mikrosensorik, Vieweg, Wiesbaden, 1996
- Völklein, F. und Zetterer, Th.: Einführung in die Mikrosystemtechnik, Vieweg, Wiesbaden, 2000

---

**Links**

**Language of instruction**
- German

**Duration (semesters)**
- 1 Semester

**Module frequency**
- annual

**Module capacity**
- unlimited

**Reference text**
- Associated with the modules:
  - Embedded Systems and Microrobotics

---

**Module level**

**Type of module**
- 1VL + 1Ü

**Previous knowledge**
- none

**Examination**
- Examination times

**Type of examination**
- Oral exam in German

**Final exam of module**
- At the end of the semester
<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3</td>
<td>WiSe</td>
<td>42</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1</td>
<td>WiSe</td>
<td>14</td>
</tr>
</tbody>
</table>

**Total module attendance time**

56 h
inf209 - Control Theory

Module label: Control Theory
Module abbreviation: inf209
Credit points: 6.0 KP
Workload: 180 h

Applicability of the module:
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik)
- Master's Programme Computing Science (Master) > Interdisziplinäre Module

Responsible persons:
- Fatikow, Sergej (module responsibility)
- Hein, Andreas (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

Prerequisites:
- Differential Equations
- Analysis II
- Fundamentals of electrical engineering

Skills to be acquired in this module:

Professional competence
The students:
- describe the core principles of steering and control of technical systems
- discuss the modelling core concepts of systems and their controllers
- name methods to determine the quality of controlled systems
- model technical systems with differential equations and their transfer functions
- develop control structures, evaluate their stability and determine their optimal control parameters

Methodological competence
The students:
- are aware of the technical challenges and solve them by including the implementations of other disciplines and methods

Social competence
The students:
- present solutions for specific questions

Self-competence
The students:
- get used to the specific challenges of the development of controlled systems

Module contents:

Basics

analog transfer elements:
- linear time invariant (LTI-) systems;
- simulation and modeling;
- step response;
- frequency response;
- frequency response locus;
- differential equations and transfer function; control loop stability;
- types of controlled systems;
- types of linear controllers;

linear control loops:
- reference and disturbance reaction of the controlled system;
- rules for control loop optimization;
- methods of analysis and synthesis, implementation;
- computer-based control MATLAB/Simulink
Recommended reading

- Unbehauen, H.: Regelungstechnik I, Klassische Verfahren zur Analyse und Synthese linearer kontinuierlicher Regelsysteme
- Lutz, H. und Wendt, W.: Taschenbuch der Regelungstechnik
- Further reading will be announced at lecture

Links

Language of instruction: German
Duration (semesters): 1 Semester
Module frequency: annual
Module capacity: unlimited
Module level:
Teaching/Learning method: 1VL + 1Ü
Previous knowledge:
- Differential Equations
- Analysis II
- Fundamentals of electrical engineering

Examination

Examination times: At the end of the lecture period
Type of examination: Hands-on exercises and written or oral exam

Type of course

<table>
<thead>
<tr>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>3</td>
<td>WiSe</td>
<td>42</td>
</tr>
<tr>
<td>Exercises</td>
<td>1</td>
<td>WiSe</td>
<td>14</td>
</tr>
</tbody>
</table>

Total module attendance time: 56 h
inf210 - Signal and Image Processing

Module label: Signal and Image Processing
Module abbreviation: inf210
Credit points: 6.0 KP
Workload: 180 h

Applicability of the module:
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlpflicht Technische Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflichtmodule (Technische Informatik)
- Master's Programme Computing Science (Master) > Interdisziplinäre Module

Responsible persons:
- Hein, Andreas (module responsibility)
- Fränzle, Martin Georg (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

Prerequisites:
Module math040 Analysis II b: Differential calculus of several variables

Skills to be acquired in this module:

Professional competence:
The students:
- name the concepts of signal and image processing in technical systems
- name the methods/algorithms of preprocessing, filtering, classification, interpretation and visualisation of signals and pictures
- Select algorithms appropriately
- evaluate the effectivenes of algorithms
- design algorithms and processing chains and evaluate their quality

Methodological competence:
The students:
- get used to specific subjects of signal and image processing

Social competence:
The students:
- present solutions for specific questions in signal and image processing

Self-competence:
The students:
- reflect their solutions by using methods learned in this course

Module contents:

Basic Concepts:
- Signal Processing
- Signal Spaces and Signal Processing Systems
- Discrete and Constant Signals
- Labelling of Signal Transmitters with Test Signals
- Representations Areas and Transformations
- Time-Discrete Systems and Scanning
- Estimation and Filtering
- Construction with MATLAB
- Image Processing

Introduction / Range of Applications:
- Functional Transformation
- Image Enhancement/Filtering
- Segmentation
- 3D Reconstruction an Visualization

Recommended reading:

Essential:
- Foliensammlung zur Vorlesung

Recommended:
- Meyer, M.; Signalverarbeitung: Analoge und digitale Signale, Systemen und Filter
Grüningen, D. C. v.; Digitale Signalverarbeitung: mit einer Einführung in die kontinuierlichen Signale und Systeme
Tönnies, K.; Grundlagen der Bildverarbeitung; Pearson Studium 2005
Lehmann, Th.; Oberschelp, W.; Pelinak, E.; Pepges, R.; Bildverarbeitung in der Medizin; Springer Verlag 1997
Handels, H.; Medizinische Bildverarbeitung; Teubner Verlag, Stuttgart
Leipzig 2000 weiterführende Literatur wird in der Vorlesung bekannt gegeben

Links
Language of instruction German
Duration (semesters) 1 Semester
Module frequency annual
Module capacity unlimited
Module level
Type of module
Teaching/Learning method 1VL + 1Ü
Previous knowledge Module math040 Analysis II b: Differential calculus of several variables
Examination Examination times Type of examination
Final exam of module At the end of the semester Hands-on exercises and written or oral exam
Type of course Comment SWS Frequency Workload of compulsory attendance
Lecture 2 WiSe 28
Exercises 2 WiSe 28
Total module attendance time 56 h
Wahlpflicht Theoretische Informatik (30 KP)

inf400 - Theoretical Computer Science: Logic

<table>
<thead>
<tr>
<th>Module label</th>
<th>Theoretical Computer Science: Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf400</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
</tbody>
</table>

Applicability of the module

- Bachelor's Programme Computing Science (Bachelor) > Basismodule
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Theoretische Informatik)
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Akzentsetzungsbereich

Responsible persons

- Olderog, Ernst-Rüdiger (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

Prerequisites

No participant requirements

Skills to be acquired in this module

Introduction to propositional logic, predicate logic, logic programming, and temporal logic

Professional competence

The students:

- know syntax, semantics and applications of propositional logic, predicate logic, logic programming, and temporal logic
- specify problems by using logical formulas
- solve questions concerning propositional formulas with truth tables
- draw conclusions in the field of propositional logic and predicate logic by means of natural deduction
- answer queries to logic programs by using SLD resolution
- perform model checking of Kripke structures with regard to CTL formulas algorithmically

Methodological competence

The students:

- recognize logic as a versatile tool in computer science

Social competence

The students:

- work together in small groups to solve problems
- present solutions to problems to groups of other students

Self-competence

The students:

- learn persistence in pursuing difficult tasks
- learn precision in writing down solutions

Module contents

The course introduces propositional, predicate and temporal logic. In computer science it is essential to have a good understanding of logic because the language of logical formulas is widely used in the field of computer science. For example, Boolean expressions appear in every programming language and in circuit design; Horn clauses are used in knowledge representation; predicate logic and temporal logic are used for specifying software and hardware. More recent applications such as interactive and automatic proving as well as the logic programming language PROLOG emphasize the tool character of logic in computer science. The course introduces syntax, semantics, procedures, and calculi to prove the validity of formulas of propositional, predicate, and temporal logic. This is illustrated by many examples. Central is the concept of logical consequence.

Topics:

- propositional logic: syntax and semantics, truth tables, natural deduction
- predicate logic: syntax and semantics, natural deduction
- logic programming: declarative and procedural semantics, unification algorithm (Robinson), SLD resolution, PROLOG
- temporal logic CTL: syntax and semantics of Kripke structures, CTL
Recommended reading

**Essential:** Script "Logik" (in German), in its current edition


---

## Links

- **Language of instruction:** German
- **Duration (semesters):** 1 Semester
- **Module frequency:** annual
- **Module capacity:** unlimited
- **Module level:**
- **Type of module:**
- **Teaching/Learning method:** 1VL + 1Ü
- **Previous knowledge:** none
- **Examination:**
  - **Examination times:**
  - **Type of examination:** written exam or oral exam
- **Final exam of module:** At the end of the lecture period
- **Type of course**

### Type of course

<table>
<thead>
<tr>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>3</td>
<td>SoSe</td>
<td>42</td>
</tr>
<tr>
<td>Exercises</td>
<td>1</td>
<td>SoSe</td>
<td>14</td>
</tr>
</tbody>
</table>

### Total module attendance time 56 h
inf401 - Foundations of Theoretical Computer Science

<table>
<thead>
<tr>
<th>Module label</th>
<th>Foundations of Theoretical Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>inf401</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
</tbody>
</table>

Applicability of the module
- Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule
- Bachelor's Programme Mathematics (Bachelor) > Nebenfachmodule
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Pflichtmodule
- Master of Education Programme (Hauptschule and Realschule) Computing Science (Master of Education) > Mastermodule
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Akzentsetzungsbereich

Responsible persons
- Olderog, Ernst-Rüdiger (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

Prerequisites
- No participant requirement

Skills to be acquired in this module

Professional competence
The students:
- know different classes of languages (e.g. regular and context-free languages)
- know automata models corresponding to the respective language classes (e.g. finite automata, pushdown automata, Turing machines)
- construct automata, Turing machines, and grammars for given tasks
- know equivalent formalisations of the concept of algorithm
- classify functions as algorithmically computable and problems as algorithmically decidable
- know and recognize undecidable problems
- evaluate the complexity of algorithms
- know problems that are solvable deterministically or nondeterministically in polynomial time

Methodological competence
The students:
- learn about the power of abstract models of computation

Social competence
The students:
- work together in small groups to solve problems
- present solutions to problems to groups of other students

Self-competence
The students:
- learn persistence in pursuing difficult tasks
- learn precision in writing down solutions

Module contents
In the first part of the course, different classes of languages are introduced (regular and context-free languages). For each class a matching automata model is presented (finite automata, pushdown automata). Various properties are proven for the introduced classes of languages and models of automata. In the second part of the course, we examine which functions are computable and which problems are decidable. To this end, the concept of algorithm is formalised. Turing machines and grammars turn out as equivalent approaches. We show that there are problems that are undecidable. Many of these problems are of practical interest. The third part of the course deals with the complexity of algorithms, i.e. how much time and space is required to solve a problem. In particular, we consider problems that are solvable in polynomial time, either deterministically or non-deterministically. These problems are classified as P and NP.

Recommended reading
Essential:
- Skript "Grundbegriffe der Theoretischen Informatik", jeweils in aktueller
Ausgabe

**Recommended:**


**Good secondary literature:**

- Hopcroft, Motwani, Ullman: "Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie", Pearson, 2002 (ein Klassiker...)

<table>
<thead>
<tr>
<th>Links</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Language of instruction</td>
<td>German</td>
</tr>
<tr>
<td>Duration (semesters)</td>
<td>1 Semester</td>
</tr>
<tr>
<td>Module frequency</td>
<td>annual</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Module level</td>
<td></td>
</tr>
<tr>
<td>Type of module</td>
<td></td>
</tr>
<tr>
<td>Teaching/Learning method</td>
<td>1VL + 1Ü</td>
</tr>
<tr>
<td>Previous knowledge</td>
<td>none</td>
</tr>
<tr>
<td>Examination</td>
<td></td>
</tr>
<tr>
<td>Examination times</td>
<td></td>
</tr>
<tr>
<td>Type of examination</td>
<td></td>
</tr>
<tr>
<td>Final exam of module</td>
<td></td>
</tr>
<tr>
<td>At the end of the lecture period</td>
<td>Written or oral exam</td>
</tr>
<tr>
<td>Type of course</td>
<td>Comment</td>
</tr>
<tr>
<td>SWS</td>
<td>Frequency</td>
</tr>
<tr>
<td>Workload of compulsory attendance</td>
<td></td>
</tr>
<tr>
<td>Lecture</td>
<td>3</td>
</tr>
<tr>
<td>Exercises</td>
<td>1</td>
</tr>
<tr>
<td>Total module attendance time</td>
<td></td>
</tr>
</tbody>
</table>
**inf407 - Program Verification**

**Module label**  
Program Verification

**Module abbreviation**  
inf407

**Credit points**  
6.0 KP

**Workload**  
180 h

**Applicability of the module**
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Theoretische Informatik)

**Responsible persons**
- Olderog, Ernst-Rüdiger (module responsibility)
- Lehrenden, Die im Modul (authorised to take exams)

**Prerequisites**
- Theoretical computer science I and II

**Skills to be acquired in this module**
Introduction to methods for proving the correctness of sequential, parallel, and distributed programs. **Professional competence**
The students:
- describe operational semantics of sequential, parallel, and distributed programs - Know the concepts of partial and total correctness of programs
- establish soundness and completeness of proof systems
- construct input-output specifications of programs
- conduct correctness proofs for programs of different classes with the help of proof rules
- check interference and deadlock freedom of parallel programs
- transform parallel and distributed programs into nondeterministic programs

**Methodological competence**
The students:
- recognize correctness as an important aspect of programs and informatics systems

**Social competence**
The students:
- work together in small groups to solve problems
- present their solutions to groups of other students

**Self-competence**
The students:
- learn persistence in pursuing difficult tasks
- learn precision in specifying problems

**Module contents**
Program verification is a systematic approach to show the absence of errors in programs. For this purpose desirable behavioural properties of a given program are proven. For instance, a sorting program should only deliver sorted arrays. Partial correctness, termination, and the absence of runtime errors are essential for sequential programs. Additional behavioural properties are of interest for parallel programs: absence of interference, absence of deadlocks, and fair behaviour. The module focuses on the verification of parallel programs. For this purpose classic methods of Hoare's logic are combined with more recent techniques of program transformation. Sequential programs are covered in preparation for this.

**Recommended reading**
**essential:**  
Or the extended English version:  

**Links**

**Language of instruction**  
German

**Duration (semesters)**  
1 Semester

**Module frequency**  
irregular

**Module capacity**  
unlimited
### Module level

<table>
<thead>
<tr>
<th>Type of module</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Teaching/Learning method</td>
<td>1VL + 1Ü</td>
</tr>
<tr>
<td>Previous knowledge</td>
<td>Theoretical computer science I and II</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Final exam of module</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Examination times</td>
<td>At the end of the lecture period</td>
</tr>
<tr>
<td>Type of examination</td>
<td>Written exam or oral exam</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of course</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload of compulsory attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3</td>
<td>SoSe oder WiSe</td>
<td>42</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1</td>
<td>SoSe oder WiSe</td>
<td>14</td>
</tr>
</tbody>
</table>

**Total module attendance time**

56 h
Abschlussmodul

bam - Bachelor Thesis and Colloquium

<table>
<thead>
<tr>
<th>Module label</th>
<th>Bachelor Thesis and Colloquium</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module abbreviation</td>
<td>bam</td>
</tr>
<tr>
<td>Credit points</td>
<td>15.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>450 h</td>
</tr>
<tr>
<td>Applicability of the module</td>
<td>Dual-Subject Bachelor’s Programme Computing Science (Bachelor) &gt; Abschlussmodul</td>
</tr>
</tbody>
</table>
| Responsible persons | Diethelm, Ira (module responsibility)  
| | Lehrenden, Die im Modul (authorised to take exams) |
| Prerequisites | No participant requirements |
| Skills to be acquired in this module | The students are able to process and write on a scientifically oriented computer science topic. |

**Professional competence**
The students:
- evaluate the possibilities and limits of computer science methods and tools and apply them appropriately

**Methodological competence**
The students:
- select appropriate methods and tools and evaluate them
- analyse problems using the latest technical and scientific literature
- implement software projects and design hardware with the latest computer science tools
- reflect a (computer) science topic under guidance, write an article (seminar paper or thesis) and present their results scientifically

**Social competence**
The students:
- recognise conflicts and solve them in a team
- use presentation and project management methods appropriately
- identify and assume responsibility for tasks
- are aware of the social impact of their computational/informatical actions, as well as the consequences of information technologies

**Self-competence**
The students:
- select priorities appropriately, also their own
- plan their computer science actions independently  
  Complement and deepen their knowledge and adapt it to the latest developments in IT independently
- evaluate their results and discuss them with users and experts

**Module contents**
A state-of-the-art computer science topic is processed theoretically, scientifically and practically. The student presents the results.

**Recommended reading**
According to the topic

**Links**
Please contact the student advisory service! Some groups publish information and forms on their web pages.

**Language of instruction**
German

**Duration (semesters)**
1 Semester

**Module frequency**
every semester

**Module capacity**
unlimited

**Module level**

**Type of module**
Teaching/Learning method 1S

**Previous knowledge**
none
<table>
<thead>
<tr>
<th>Examination</th>
<th>Examination times</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td>varying</td>
<td>thesis, seminar paper</td>
</tr>
</tbody>
</table>

**Type of course**  
Seminar

**SWS**  
2

**Frequency**  
SoSe und WiSe

**On-site workload**  
28 h