Basismodule

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 code</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>
<tr>
<td>Used in course of study</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fach-Bachelor Informatik &gt; Basismodule</td>
</tr>
<tr>
<td></td>
<td>Fach-Bachelor Mathematik &gt; Nebenfachmodule</td>
</tr>
<tr>
<td></td>
<td>Fach-Bachelor Wirtschaftsinformatik &gt; Basismodule</td>
</tr>
<tr>
<td></td>
<td>Fach-Bachelor Wirtschaftswissenschaften &gt; Studienrichtung Wirtschaftsinformatik</td>
</tr>
<tr>
<td></td>
<td>Zwei-Fächer-Bachelor Informatik &gt; Basismodule</td>
</tr>
<tr>
<td>Contact person</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Module responsibility</td>
</tr>
<tr>
<td></td>
<td>Andreas Winter</td>
</tr>
<tr>
<td></td>
<td>Dietrich Boles</td>
</tr>
<tr>
<td>Authorized examiners</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Andreas Winter</td>
</tr>
<tr>
<td></td>
<td>Dietrich Boles</td>
</tr>
<tr>
<td></td>
<td>Die im Modul Lehrenden</td>
</tr>
<tr>
<td>Entry requirements</td>
<td></td>
</tr>
<tr>
<td>Skills to be acquired in this module</td>
<td></td>
</tr>
<tr>
<td>Module contents</td>
<td></td>
</tr>
<tr>
<td>Reader's advisory</td>
<td></td>
</tr>
<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></td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Moduleart</td>
<td>---</td>
</tr>
<tr>
<td>Lern-/Lehrform / Type of program</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
<td></td>
</tr>
<tr>
<td>Examination</td>
<td>Time of examination</td>
</tr>
<tr>
<td>Final exam of module</td>
<td>Type of examination</td>
</tr>
<tr>
<td>Comment</td>
<td>SWS</td>
</tr>
<tr>
<td>Lecture</td>
<td>4.00</td>
</tr>
<tr>
<td>Exercises</td>
<td>2.00</td>
</tr>
<tr>
<td>Total time of attendance for the module</td>
<td></td>
</tr>
</tbody>
</table>
inf200 - Computer Engineering I

Module label
Computer Engineering I

Module code
inf200

Credit points
6.0 KP

Workload
180 h

Used in course of study
- Fach-Bachelor Informatik > Basismodule
- Fach-Bachelor Mathematik > Nebenfachmodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person
Module responsibility
- Wolfgang Nebel
  - Die im Modul Lehrenden

Authorized examiners
- Wolfgang Nebel
  - Die im Modul Lehrenden

Entry requirements
The participants 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 competence
The students:
- identify the fundamental components of digital circuitry and digital computers,
- are aware of the virtues of hierarchical and abstract descriptions of hardware systems,
- name the fundamental parameters, criteria, conditions, and development trends of current and future hardware design
- describe the basic concepts of current computer architectures and the execution of machine programs

Methodological competence
The students:
- evaluate computer architectures
- design and optimize digital hardware components
- transfer systematic methods of hardware design to unknown design problems

Social competence
The students:
- present their understanding of the operational principles underlying digital computers to others

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 and fundamental methods for the specification, construction and optimization of computer components to elementary components.

Reader's advisory
- handout manuscript of the course
- Patterson, D.A.; Hennessy, J.L. (1997): Computer Organization and Design:
  - The Hardware/Software Interface; 2. Edition; Morgan Kaufmann Publishers.

Links

Language of instruction
German

Duration (semesters)
1 Semester
<table>
<thead>
<tr>
<th>Module frequency</th>
<th>jährlich</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Modullevel</td>
<td>BC (Basiscurriculum)</td>
</tr>
<tr>
<td>Modulart</td>
<td>Pflicht</td>
</tr>
</tbody>
</table>

**Lern-/Lehrform / Type of program**

**Vorkenntnisse / Previous knowledge**

<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</th>
<th>Type of examination</th>
</tr>
</thead>
</table>

**Final exam of module**

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3.00</td>
<td>WiSe</td>
<td>42 h</td>
</tr>
<tr>
<td>Exercise or tutorial</td>
<td></td>
<td>1.00</td>
<td>WiSe</td>
<td>14 h</td>
</tr>
</tbody>
</table>

**Total time of attendance for the module**  56 h
inf201 - Computer Engineering II

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

Used in course of study:
- Fach-Bachelor Informatik > Aufbaumodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person:
- Module responsibility
  - Wolfgang Nebel
- Authorized examiners
  - Wolfgang Nebel
  - Die im Modul Lehrenden

Entry requirements

Skills to be acquired in this module:
The module qualifies students to analyse computer architectures, understand computer components, design and optimize computers and components, and to discuss domain-specific hardware design.

Professional competence:
The students:
- describe computer components
- design and optimise computer components
- understand manufacturing processes for VLSI circuits

Methodological competence:
The students:
- analyse computer architectures

Social competence:
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-competence:
The students:
- are able to assess their own competences in relation to qualified personnel from related domains

Module contents:
This module is the second part of the introduction to technical computer science. Typical examples of combinatory circuits, like an adder, are used to illustrate modular design methods. More advanced design methods are demonstrated on sequential circuits, i.e. circuits with memory. Additionally in this part, the electrotechnical fundamentals of computing are taught. The construction and the manufacturing process of digital components is explained and the scope of the introduction to computer architecture is broadened to cover embedded systems as well.

Reader's advisory:
- Lecture notes
- Oberschelp, W., Vossen, G.: Rechneraufbau und Rechnerstrukturen; Oldenbourg Verlag

Additional literature will be mentioned in the lectures

Links:

Language of instruction: German
Duration (semesters): 1 Semester
Module frequency: jährlich
<table>
<thead>
<tr>
<th>Module capacity</th>
<th>unlimited</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modullevel</td>
<td>---</td>
</tr>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
</tbody>
</table>

**Lern-/Lehrform / Type of program**

**Vorkenntnisse / Previous knowledge**

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

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>3.00</td>
<td>SuSe</td>
<td></td>
<td>42 h</td>
</tr>
<tr>
<td>Exercise or tutorial</td>
<td>1.00</td>
<td>SuSe</td>
<td></td>
<td>14 h</td>
</tr>
</tbody>
</table>

**Total time of attendance for the module**

56 h
# inf203 - Embedded Systems I

<table>
<thead>
<tr>
<th>Module label</th>
<th>Embedded Systems I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>inf203</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
<tr>
<td>Used in course of study</td>
<td>• Fach-Bachelor Informatik &gt; Akzentsetzungsmodule</td>
</tr>
<tr>
<td></td>
<td>• Master of Education (Gymnasium) Informatik &gt; Mastermodule</td>
</tr>
<tr>
<td></td>
<td>• Zwei-Fächer-Bachelor Informatik &gt; Basismodule</td>
</tr>
<tr>
<td>Contact person</td>
<td>Module responsibility</td>
</tr>
<tr>
<td></td>
<td>• Wolfgang Nebel</td>
</tr>
<tr>
<td></td>
<td>• Martin Georg Fränzle</td>
</tr>
<tr>
<td>Authorized examiners</td>
<td>• Wolfgang Nebel</td>
</tr>
<tr>
<td></td>
<td>• Martin Georg Fränzle</td>
</tr>
<tr>
<td></td>
<td>• Die im Modul Lehrenden</td>
</tr>
</tbody>
</table>

## Entry requirements

This module provides an introduction to the design of digital embedded systems.

### 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 StateCharts) 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.

## Reader's advisory

Slides and:


Secondary literature:


Links

Language of instruction
German

Duration (semesters)
1 Semester

Module frequency
jährlich

Module capacity
unlimited

Reference text
This module is compulsory for students who are specialising in "Eingebettete Systeme und Mikrorobotik".

Assicates with the modules:
In the module "Eingebettete Systeme II" additional relevant topics such as design processes, HW/SW-Partitioning, High-Level-Synthesis and Hardware discription 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".

Modullevel
---

Modultyp
je nach Studiengang Pflicht oder Wahlpflicht

Lern-Lehrform / Type of program
- Grundlagen der technischen Informatik
- Technische Informatik

Vorkenntnisse / Previous knowledge

Examination
Time of examination
Type of examination

Final exam of module
At the end of the semester
Written or oral exam

Course type
Comment
SWS
Frequency
Workload attendance

Lecture
3.00
WiSe
42 h

Exercises
1.00
WiSe
14 h

Total time of attendance for the module
56 h
inf204 - Embedded Systems II

Module label: Embedded Systems II
Module code: inf204
Credit points: 6.0 KP
Workload: 180 h

Used in course of study:
- Fach-Bachelor Informatik > Akzentsetzungsmodul
- Master of Education (Gymnasium) Informatik > Mastermodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person:
Module responsibility
- Wolfgang Nebel
- Martin Georg Fränzle

Authorized examiners:
- Wolfgang Nebel
- Martin Georg Fränzle
- Die im Modul Lehrenden

Entry requirements:
Skills to be acquired in this module:

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.

Reader's advisory


Secondary literature:


Links

Language of instruction: German

Duration (semesters): 1 Semester

Module frequency: jährlich

Module capacity: unlimited

Reference text: This module is supposed to be a compulsory module for students who are specialising in "Eingebettete Systeme und Mikrorobotik".

Modullevel: ---

Modulart: je nach Studiengang Pflicht oder Wahlpflicht

Lern-/Lehrform / Type of program

Vorkenntnisse / Previous knowledge

Examination

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

Course type

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>3.00</td>
<td>SuSe</td>
<td>42 h</td>
<td></td>
</tr>
<tr>
<td>Exercises</td>
<td>1.00</td>
<td>SuSe</td>
<td>14 h</td>
<td></td>
</tr>
</tbody>
</table>

Total time of attendance for the module: 56 h
inf205 - Formal Methods in Embedded System Design

Module label
Formal Methods in Embedded System Design

Module code
inf205

Credit points
6.0 KP

Workload
180 h

Used in course of study
• Fach-Bachelor Informatik > Akzentsetzungsmodule
• Master of Education (Gymnasium) Informatik > Mastermodule
• Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person
Module responsibility
• Martin Georg Fränzle

Authorized examiners
• Martin Georg Fränzle
• Die im Modul Lehrenden

Entry requirements

Skills to be acquired in this module
Embedded computer systems sustain a permanent interaction with their environment. This interaction may lead to hardly predictable stimuli and reponse sequences, which complicates the design and validation of such systems tremendously. As in more mature engineering disciplines, formal analytical models have been proposed as a remedy. Their role in the design flow is equivalent to the use of structural analysis and material science within, e.g., building statics. Pertinent formal methods for and formal models of embedded systems cover, for instance, execution time, power demand, and possible system dynamics. As they represent relevant aspects of a system in a formal, mathematical way, they often permit automatic analysis - i.e., to derive characteristic data - and automatic certificate generation. The distinguishing factor to more traditional forms of analysis like testing and profiling is the exhaustive form of analysis achieved by mathematical methods, which guarantee that the results apply for any environmental interaction. This is in stark contrast to the inherently incomplete coverage provided by test-based methods.

The lectures explain a series of increasingly more expressive formal models and the related automatic analysis techniques. The exercise classes complement these theoretical insights by hands-on experience with state of the art formal analysis tools and offer the possibility to build such tools oneself.

Professional competence
The students:
• Evaluate the consequences of certificates applied by formal methods
• Evaluate the suitability of available verification tools for a partial aspect and system class
• Use these tools and interpret their results and improve the examined system
• Prepare system models for automatic analysis methods and abstract or encode the systems symbolically (or otherwise) accordingly
• Design and implement verification algorithms

Methodological competence
The students:
• Are able to model complex and heterogeneous systems by adequate mathematical modelling techniques
• Know pertinent mathematical models for system dynamics and are able to transfer them to other problem domains.

Social competence
The students:
• Develop and implement fundamental verification algorithms in teams
• Discuss the relative merits of alternative algorithms and formalisms

Self-competence
The students:

Module contents
The module explains semantic models for reactive, real-time, and hybrid discrete-continuous systems and gives examples for pertinent specification logics. It gradually develops state-exploratory verification algorithms, both of explicit-state and symbolic shape, as relevant to the development of reliable hardware and software systems.

The lectures present the semantic, logical, and algorithmic foundations of the automatic analysis for embedded software systems. The exercise classes complement this by providing space for experimenting with formalisms and tools in teams. The second half of the semester is dedicated to the semester project, which either deals...
with implementing an automatic verifier or with in-depth usage of existing tools on examples of industrially relevant size

Reader's advisory


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>jährlich</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Modullevel</td>
<td>AS (Akzentsetzung)</td>
</tr>
<tr>
<td>Modulart</td>
<td>Wahlpflicht</td>
</tr>
<tr>
<td>Lern-/Lehrform / Type of program</td>
<td>V+Ü</td>
</tr>
</tbody>
</table>

Vorkenntnisse / Previous knowledge

<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</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>Semester project</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3.00</td>
<td>WiSe</td>
<td>42 h</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1.00</td>
<td>WiSe</td>
<td>14 h</td>
</tr>
</tbody>
</table>

Total time of attendance for the module 56 h
inf207 - Electrical Engineering

<table>
<thead>
<tr>
<th>Module label</th>
<th>Electrical Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</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>
| Used in course of study | • Fach-Bachelor Informatik > Akzentsetzungsmodule  
|                    | • Master of Education (Gymnasium) Informatik > Mastermodule  
|                    | • Zwei-Fächer-Bachelor Informatik > Basismodule            |
| Contact person     | Module responsibility                                      |
|                    | • Andreas Hein                                             |
|Authorized examiners| • Andreas Hein  
|                    | • Die im Modul Lehrenden                                   |

Entry requirements

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

Reader's advisory

essential:

- slides

recommended:

### 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>jährlich</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Modullevel</td>
<td>---</td>
</tr>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
</tbody>
</table>

### Lern-/Lehrform / Type of program

| V & Ü |

### Vorkenntnisse / Previous knowledge

### Final exam of module

<table>
<thead>
<tr>
<th>Course type</th>
<th>Time of examination</th>
<th>Type of examination</th>
<th>Hands-on exercises / written exam or oral exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>At the End of the Semester</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.00</td>
<td>SuSe</td>
<td>42 h</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>SuSe</td>
<td>14 h</td>
</tr>
</tbody>
</table>

### Total time of attendance for the module

| 56 h |
inf208 - Microrobotics and Microsystems Technology

Module label: Microrobotics and Microsystems Technology
Module code: inf208
Credit points: 6.0 KP
Workload: 180 h
Used in course of study:
- Fach-Bachelor Informatik > Akzentsetzungsmodule
- Master of Education (Gymnasium) Informatik > Mastermodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person
- Module responsibility: Sergej Fatikow
- Authorized examiners:
  - Sergej Fatikow
  - Die im Modul Lehrenden

Entry 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, electro rheological 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

Reader's advisory
Essential:
Lecture notes

**Recommended:**


**Secondary Literature (only available for some subareas):**

- Elbel, Th.: Mikrosensorik, Vieweg, Wiesbaden, 1996
- Völklein, F. und Zetterer, Th.: Einführung in die Mikrosystemtechnik, Vieweg, Wiesbaden, 2000

<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>
<tr>
<td>Reference text</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Modulelevel</td>
</tr>
<tr>
<td>Modulart</td>
</tr>
</tbody>
</table>

**Lern-/Lehrform / Type of program**

<table>
<thead>
<tr>
<th>Vorkenntnisse / Previous knowledge</th>
<th>Analysis II or Numerics</th>
</tr>
</thead>
</table>

**Examination**

<table>
<thead>
<tr>
<th>Final exam of module</th>
<th>At the end of the semester</th>
<th>Oral exam in German</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3.00</td>
<td>WiSe</td>
<td>42 h</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1.00</td>
<td>WiSe</td>
<td>14 h</td>
</tr>
</tbody>
</table>

**Total time of attendance for the module**

|               | 56 h   |
inf209 - Control Theory

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

Used in course of study:
- Fach-Bachelor Informatik > Akzentsetzungsmodule
- Master of Education (Gymnasium) Informatik > Mastermodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person:
Module responsibility
- Sergej Fatikow
- Andreas Hein

Authorized examiners
- Sergej Fatikow
- Andreas Hein
- Die im Modul Lehrenden

Entry requirements:
- Module Differential Equations
- Module Basics Electrical Engineering

Skills to be acquired in this module:
Instruction on theoretical and mathematical basics of control engineering

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 modelling; 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

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

Links
<table>
<thead>
<tr>
<th><strong>Language of instruction</strong></th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Duration (semesters)</strong></td>
<td>1 Semester</td>
</tr>
<tr>
<td><strong>Module frequency</strong></td>
<td>jährlich</td>
</tr>
<tr>
<td><strong>Module capacity</strong></td>
<td>unlimited</td>
</tr>
<tr>
<td><strong>Modullevel</strong></td>
<td>---</td>
</tr>
<tr>
<td><strong>Modulart</strong></td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
</tbody>
</table>

**Lern-/Lehrform / Type of program**

**Vorkenntnisse / Previous knowledge**

<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Final exam of module</strong></td>
<td>At the end of the lecture period</td>
<td>Hands-on exercises and written or oral exam</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>3.00</td>
<td>WiSe</td>
<td></td>
<td>42 h</td>
</tr>
<tr>
<td>Exercises</td>
<td>1.00</td>
<td>WiSe</td>
<td></td>
<td>14 h</td>
</tr>
</tbody>
</table>

**Total time of attendance for the module** 56 h
inf210 - Signal and Image Processing

Module label | Signal and Image Processing
Module code | inf210
Credit points | 6.0 KP
Workload | 180 h

Used in course of study
- Fach-Bachelor Informatik > Akzentsetzungsmodul
- Master of Education (Gymnasium) Informatik > Mastermodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person
Module responsibility
- Martin Georg Fränzle
- Andreas Hein

Authorized examiners
- Martin Georg Fränzle
- Andreas Hein
- Die im Modul Lehrenden

Entry requirements

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 effectiveness 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

Reader's advisory
essential:
Slides
recommended:
- Meyer, M.; Signalverarbeitung: Analoge und digitale Signale, Systeme 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.; Peges, R.; Bildverarbeitung in der Medizin; Springer Verlag 1997
- Handels, H.; Medizinische Bildverarbeitung; Teubner Verlag, Stuttgart - Leipzig 2000

<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>jährlich</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Modular</td>
<td>---</td>
</tr>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
<tr>
<td>Lern-/Lehrform / Type of program</td>
<td>V &amp; Ü</td>
</tr>
</tbody>
</table>
| Vorkenntnisse / Previous knowledge | - Differenzialgleichungen  
- Analysis II |

<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</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 and written or oral exam</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>2.00</td>
<td>SuSe</td>
<td>28 h</td>
<td></td>
</tr>
<tr>
<td>Exercises</td>
<td>2.00</td>
<td>SuSe</td>
<td>28 h</td>
<td></td>
</tr>
</tbody>
</table>

Total time of attendance for the module 56 h
inf400 - Theoretical Computer Science I

<table>
<thead>
<tr>
<th>Module label</th>
<th>Theoretical Computer Science I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</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>
| Used in course of study           | • Fach-Bachelor Informatik > Basismodule  
|                                   | • Zwei-Fächer-Bachelor Informatik > Basismodule  |
| Contact person                    | Module responsibility          |
|                                   | • Eike Best                    |
|                                   | • Annegret Habel               |
|                                   | • Ernst-Rüdiger Olderog        |
| Authorized examiners              | • Eike Best                    |
|                                   | • Annegret Habel               |
|                                   | • Ernst-Rüdiger Olderog        |
|                                   | • Die im Modul Lehrenden       |
| Entry requirements                | Entry 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 model checking algorithm

**Reader's advisory**

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

**Recommended:**

**Good secondary reading:**

**Links**

<table>
<thead>
<tr>
<th>Language of instruction</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Duration (semesters)</strong></td>
<td>1 Semester</td>
</tr>
<tr>
<td><strong>Module frequency</strong></td>
<td>jährlich</td>
</tr>
<tr>
<td><strong>Module capacity</strong></td>
<td>unlimited</td>
</tr>
<tr>
<td><strong>Modullevel</strong></td>
<td>BC (Basisscourriculum)</td>
</tr>
<tr>
<td><strong>Modulart</strong></td>
<td>Pflicht</td>
</tr>
<tr>
<td><strong>Lern-/Lehrform / Type of program</strong></td>
<td>VL + Ü</td>
</tr>
</tbody>
</table>

**Vorkenntnisse / Previous knowledge**

**Final exam of module**

<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</th>
<th>Type of examination</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course type</td>
<td>Comment</td>
<td>SWS</td>
<td>Frequency</td>
</tr>
<tr>
<td>Lecture</td>
<td></td>
<td>3.00</td>
<td>SuSe</td>
</tr>
<tr>
<td>Exercise or tutorial</td>
<td></td>
<td>1.00</td>
<td>SuSe</td>
</tr>
</tbody>
</table>

**Total time of attendance for the module**

| Total time of attendance for the module | 56 h |
inf401 - Theoretical Computer Science II

Module label: Theoretical Computer Science II

Module code: inf401

Credit points: 6.0 KP

Workload: 180 h

Used in course of study:
- Fach-Bachelor Informatik > Aufbaumodule
- Fach-Bachelor Mathematik > Nebenfachmodule
- Zweifächer-Bachelor Informatik > Basismodule

Contact person:
Module responsibility:
- Eike Best
- Annegret Habel
- Ernst-Rüdiger Olderog

Authorized examiners:
- Eike Best
- Annegret Habel
- Ernst-Rüdiger Olderog
- Die im Modul Lehrenden

Entry requirements:

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.
**Reader's advisory**
- essentiell: Skript "Grundbegriffe der Theoretischen Informatik", jeweils in aktueller Ausgabe
- Gute Sekundärliteratur: Hopcroft, Motwani, Ullman: "Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie", Pearson, 2002 (ein Klassiker...)

**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>jährlich</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Modullevel</td>
<td>---</td>
</tr>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
<tr>
<td>Lern-/Lehrform / Type of program</td>
<td>V + Ü</td>
</tr>
</tbody>
</table>

**Vorkenntnisse / Previous knowledge**

<table>
<thead>
<tr>
<th>Final exam of module</th>
<th>Time of examination</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course type</td>
<td>Comment</td>
<td>SWS</td>
</tr>
<tr>
<td>Lecture</td>
<td>3.00</td>
<td>WiSe</td>
</tr>
<tr>
<td>Exercise or tutorial</td>
<td>1.00</td>
<td>WiSe</td>
</tr>
</tbody>
</table>

**Total time of attendance for the module**

56 h
inf402 - Graph Transformation Systems

Module label: Graph Transformation Systems

Module code: inf402

Credit points: 6.0 KP

Workload: 180 h

Used in course of study:
- Fach-Bachelor Informatik > Akzentsetzungsmodule
- Master of Education (Gymnasium) Informatik > Mastermodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person:
Module responsibility
- Annegret Habel

Authorized examiners
- Annegret Habel
- Die im Modul Lehrenden

Entry requirements
Skills to be acquired in this module:

Professional competence
- The students:
  - Know the basics of graph transformation systems and graph programs
  - Describe graph transformation systems and graph programs
  - Define the Turing completeness of graph programs
  - Model systems and system changes
  - Prove sequential and parallel independence of derivations
  - Prove termination and confluence of graph transformation systems

Methodological competence
- The students:
  - Recognize graph transformation systems as a versatile tool for modelling 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:
Graphs are practically used in all areas of computer science to display complex structures. Some examples are flow charts, circuit diagrams, record structures, parse trees and functional and logical expressions. Such structures can be dynamically changed by graph rewriting systems. The changing process is represented by rewriting rules. This module gives an introduction to the field of graph transformation systems. It deals with reversibility, embedding and restriction of derivations, sequential and parallel independency, termination and confluence.

Reader's advisory:

Links:
Language of instruction: German
Duration (semesters): 1 Semester
Module frequency: im 2-Jahres-Zyklus
<table>
<thead>
<tr>
<th>Module capacity</th>
<th>unlimited</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modullevel</td>
<td>---</td>
</tr>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
<tr>
<td>Lern-/Lehrform / Type of program</td>
<td>V &amp; Ü</td>
</tr>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
<td>Theoretische Informatik II</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3.00</td>
<td>SuSe or WiSe</td>
<td>42 h</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1.00</td>
<td>SuSe or WiSe</td>
<td>14 h</td>
</tr>
</tbody>
</table>

**Total time of attendance for the module**: 56 h
**inf403 - Cryptology**

<table>
<thead>
<tr>
<th>Module label</th>
<th>Cryptology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>inf403</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
</tbody>
</table>
| Used in course of study | • Fach-Bachelor Informatik > Akzentsetzungsmodule  
• Master of Education (Gymnasium) Informatik > Mastermodule  
• Zwei-Fächer-Bachelor Informatik > Basismodule |

**Contact person**
- Module responsibility
  - Elke Wilkeit
- Authorized examiners
  - Elke Wilkeit
  - Die im Modul Lehrenden

**Entry requirements**

**Skills to be acquired in this module**

Cryptology is a key technology for the security of worldwide computer nets. Modern cryptographic techniques are used to keep data secret, sign electronic messages, control computer network access, secure electronic financial transactions, protect copyrights, among others. In view of these applications users should be able to assess the efficiency and security of these key technologies. For this purpose, it is important not only to know the function of cryptographic processes, it is also important to understand their mathematical basics. Both is explained in this module.

**Professional competence**
The students:
- identify basic concepts of cryptography and explain them by examples
- know relevant cryptosystems, apply them and assess their security
- are familiar in using mathematical basics of cryptographic algorithms
- implement cryptographic algorithms and prove their correctness and estimations of their complexity

**Methodological competence**
The students:
- assess the efficiency and security of cryptographic processes
- extend their knowledge about algorithms and their complexity
- develop their implementation skills in particular the handling of very large numbers
- analyze simple encryption using well-known and own techniques

**Social competence**
The students
- use the language of mathematics to discuss in groups with different knowledge about problems
- present their ideas in an understandable way
- expand and improve their own ideas through the proposals of their fellow students

**Self-competence**
The students:
- reflect their knowledge about security in IT systems
- reflect their knowledge about algorithms and their complexity
- experience the development of a new field of knowledge within a short amount of time
- discover new applications of mathematical contexts

**Module contents**

| A) Mathematical Basics: Integers; Polynomials; Congruences; Residue Class Rings |
| B) Encryption |
| C) Probability and Perfect Security |
| D) Symmetric Encryption (DES, AES) |
| E) Generation of Prime Numbers |
| F) Public-Key-Encryption |
| G) Factorisation and Discrete Logarithms |
| H) Cryptographic Hash Functions and Digital Signatures |
| I) Identification and Certification |

**Reader's advisory**

Lecture notes; further literature will be announced in the lecture.

For attunement: Singh, Simon: The Code Book: Science of Secrecy from Ancient Egypt to Quantum

<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>jährlich</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Modullevel</td>
<td>---</td>
</tr>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lern-/Lehreform / Type of program</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
<td>Mathematical and computer science basics</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td>At the end of the lecture periode</td>
<td>Written exam</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>2.00</td>
<td>WiSe</td>
<td>28 h</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>2.00</td>
<td>WiSe</td>
<td>28 h</td>
</tr>
</tbody>
</table>

| Total time of attendance for the module | 56 h |

| Workload attendance | 27 / 76 |
The behaviour of modern, highly parallel, digital systems may be extremely complex. Graphical and algorithmic support may be very valuable in facilitating their design, construction, and analysis. Petri nets are a basic, widely used graphical model for the specification of parallel systems. They also provide and support a range of flexible algorithmic methods for the analysis of such systems. This module teaches the basic theory and applications of Petri nets, for the purpose of specifying and visualising, as well as for constructing and analysing highly parallel systems.

**Professional competence**
The students:

- define basic concepts of Petri nets
- classify Petri nets according to their salient properties
- analyse and synthesise Petri nets
- apply Petri nets in the context of well-defined problems

**Methodological competence**
The students:

- can apply specification and analysis methods based on Petri nets

**Social competence**
The students:

- present solutions to given problems to a wider audience

**Module contents**

- Basic concepts of Petri net theory.
- Petri net languages.
- Reachability and coverability.
- Marking equation.
- Linear-algebraic and graph-theoretic structure of Petri nets.
- Free-choice nets.
- Program verification using traps.
- Computing functions with nets.
- Untoldings.
- High-level nets.

**Reader's advisory**


**Links**

**Languages of instruction**

German, English

**Duration (semesters)**

1 Semester
<table>
<thead>
<tr>
<th>Module frequency</th>
<th>jährlich</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Module level</td>
<td>---</td>
</tr>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
<tr>
<td>Lern-/Lehrform / Type of program</td>
<td></td>
</tr>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</th>
<th>Type of examination</th>
<th>Written or oral exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td>At the end of the lecture period</td>
<td>Frequency</td>
<td>Workload attendance</td>
</tr>
<tr>
<td>Course type</td>
<td>Comment</td>
<td>SWS</td>
<td></td>
</tr>
<tr>
<td>Lecture</td>
<td>2.00</td>
<td>SuSe</td>
<td></td>
</tr>
<tr>
<td>Exercises</td>
<td>2.00</td>
<td>SuSe</td>
<td></td>
</tr>
<tr>
<td>Total time of attendance for the module</td>
<td>56 h</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
inf405 - Algorithmic Graph Theory

Module label: Algorithmic Graph Theory
Module code: inf405
Credit points: 6.0 KP
Workload: 180 h

Used in course of study:
- Fach-Bachelor Informatik > Akzentsetzungsmodule
- Master of Education (Gymnasium) Informatik > Mastermodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person:
- Module responsibility: Elke Wilkeit
- Authorized examiners: Elke Wilkeit, Die im Modul Lehrenden

Entry requirements:
Graphs are the most frequently used abstraction in computer science. Every system which consists of discrete states or objects and relations between these can be modelled as a graph. Most applications require efficient algorithms to process such graphs (Turau, 1996). This module provides typical graph theory problems and algorithmic solutions. They are discussed with regard to their efficiency and applicability and many of the algorithms will be implemented. An important aspect of this module is to consider different approaches to problems and learn different solution strategies.

Professional competence
The students:
- identify basic terms of graph theory and optimization and illustrate them with examples
- name typical graph theory problems and algorithmic solutions
- identify situations where graph algorithms can be applied
- discuss typical graph theory problems and algorithmic solutions with regard to their efficiency and applicability.
- implement graph algorithms
- know proof strategies and are able to apply them

Methodological competence
The students:
- extend their knowledge about algorithms and their complexity
- develop their programming skills
- expand their range of methods of mathematical modelling

Social competence
The students:
- use the language of mathematics to discuss problems in groups with different knowledge levels
- present their ideas in a comprehensible way
- Expand and improve their own ideas through the comments of their fellow students

Self-competence
The students:
- reflect their knowledge about algorithms and their complexity
- develop appropriate solutions for given problems
- challenge methods of resolution

Module contents:
A) Trees
B) Search Algorithms
C) Graph Coloring
D) Flows in Networks
E) Applications of Network Algorithms
F) Shortest Paths
G) Approximation Algorithms

Reader's advisory:
A detailed bibliography is contained in the lecture notes of this module.

<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>jährlich</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Modullevel</td>
<td>---</td>
</tr>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lern-/Lehrform / Type of program</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
<td>Mathematical and computer science basics</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Final exam of module</th>
<th>At the end of the lecture period</th>
<th>Written exam</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Course type</td>
<td>Time of examination</td>
<td>Type of examination</td>
<td>Workload attendance</td>
</tr>
<tr>
<td>Lecture</td>
<td>3.00</td>
<td>SuSe</td>
<td>42 h</td>
</tr>
<tr>
<td>Exercises</td>
<td>1.00</td>
<td>SuSe</td>
<td>14 h</td>
</tr>
</tbody>
</table>

| Total time of attendance for the module | 56 h |
inf407 - Program Verification

Module label  Program Verification
Module code    inf407
Credit points  6.0 KP
Workload       180 h

Used in course of study
- Fach-Bachelor Informatik > Akzentsetzungsmodule
- Master of Education (Gymnasium) Informatik > Mastermodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person
Module responsibility
- Ernst-Rüdiger Olderog
Authorized examiners
- Ernst-Rüdiger Olderog
- Die im Modul Lehrenden

Entry requirements

Skills to be acquired in this module

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.

Reader's advisory
essential:

Or the extended English version:
<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>
<tr>
<td>Modullevel</td>
</tr>
<tr>
<td>Modulart</td>
</tr>
<tr>
<td>Lern-Lehrform / Type of program</td>
</tr>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td>At the end of the lecture period</td>
<td>Written exam or oral exam</td>
</tr>
<tr>
<td>Course type</td>
<td>Comment</td>
<td>SWS</td>
</tr>
<tr>
<td>Lecture</td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1.00</td>
</tr>
<tr>
<td>Total time of attendance for the module</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
inf408 - Algorithms for Software Verification

Module label: Algorithms for Software Verification

Module code: inf408

Credit points: 6.0 KP

Workload: 180 h

Used in course of study:
- Fach-Bachelor Informatik > Akzentsetzungsmodulle
- Master of Education (Gymnasium) Informatik > Mastermodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person:

Module responsibility:
- Ernst-Rüdiger Olderog

Authorized examiners:
- Ernst-Rüdiger Olderog
- Die im Modul Lehrenden

Entry requirements:

Skills to be acquired in this module:

Algorithms are presented that enables an automatic analysis and verification of complex structures as used in software systems. In the exercises these algorithms will be implemented and applied to case studies.

Professional competence:
The students:
- conduct CTL model checking using examples
- construct abstract Kripke structures on the basis of given data abstractions and apply abstraction refinement to examples
- characterise the concepts of simulation and bisimulation
- understand the concept of data and transition abstraction
- describe model checking methods as instances of fixed-point algorithms

Methodological competence:
The students:
- specify reactive systems by means of Kripke structures and CTL formulas
- implement model checking methods using Java

Social competence:
The students:
- work in small groups

Self-competence:
The students:
- reflect their actions and use newly learned methods

Module contents:

Software systems consist of complex data and control structures and growing state spaces, which makes testing their correctness difficult. The big challenge for computer science is the development of automatic methods to analyse and verify software systems’ properties. In this course, algorithms for program analysis and model checking are presented and applied. The algorithms process transition systems generated from software and use abstraction techniques for data and transitions to make the state spaces analysable.

Topics:
Kripke structures, transition systems, temporal logic CTL and CTL*, fixed-point algorithms for recursive CTL-operators, model checking algorithms for CTL, simulation and bisimulation of Kripke structures, theorems on the preservation of properties under (bi-) simulations, existential und universal abstraction of Kripke structures, counterexample-guided abstraction refinement (CEGAR method)

Reader’s advisory:

<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>
<tr>
<td>Modullevel</td>
</tr>
<tr>
<td>Modulart</td>
</tr>
<tr>
<td>Lern-/Lehrform / Type of program</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vorkenntnisse / Previous knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examination</td>
</tr>
<tr>
<td>Time of examination</td>
</tr>
<tr>
<td>Type of examination</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Final exam of module</th>
</tr>
</thead>
<tbody>
<tr>
<td>First week of lecture-free period</td>
</tr>
<tr>
<td>Written exam or oral exam</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>2.00</td>
<td></td>
<td>28 h</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>2.00</td>
<td></td>
<td>28 h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total time of attendance for the module</th>
<th>56 h</th>
</tr>
</thead>
</table>
inf409 - Formal Languages

Module label: Formal Languages
Module code: inf409
Credit points: 6.0 KP
Workload: 180 h

Used in course of study:
- Fach-Bachelor Informatik > Akzentsetzungsmodul
- Master of Education (Gymnasium) Informatik > Mastermodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person
Module responsibility:
- Annegret Habel
Authorized examiners:
- Annegret Habel
- Die im Modul Lehrenden

Entry requirements
Skills to be acquired in this module

Professional competence
The students:
- Know the fundamentals of syntactic analysis and compiler construction
- Describe the complexity of fundamental syntactic analysis algorithms
- Construct no-left-recursive-grammars and grammars in normal form
- Test LL(k) and LR(k) characteristics of context-free grammars
- Construct LL(k)-Parsing and LR(k)-Parsing-Action and GOTO tables
- Apply basic syntax analysis algorithms

Methodological competence
The students:
- Perceive syntax analysis algorithms as a essential tool in computer science

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 writing down solutions

Module contents
The course introduces the fundamentals of syntax analysis and considers backtrack parsing (Top-Down & Bottom-Up Backtracking), tabular parsing methods (Cocke-Younger-Kasami & Earley) und One-Pass No Backtrack Parsing (LL(k) und LR(k)).

Reader's advisory

Links
Language of instruction: German
Duration (semesters): 1 Semester
Module frequency: im 2-Jahres-Zyklus
Module capacity: unlimited
<table>
<thead>
<tr>
<th>Modullevel</th>
<th>---</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
<tr>
<td>Lern-Lehrform / Type of program</td>
<td>V &amp; Ü</td>
</tr>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
<td>Theoretische Informatik II</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>2.00</td>
<td>SuSe</td>
<td>28 h</td>
<td>56 h</td>
</tr>
<tr>
<td>Exercises</td>
<td>2.00</td>
<td>SuSe</td>
<td>28 h</td>
<td></td>
</tr>
</tbody>
</table>

Total time of attendance for the module 56 h
inf700 - Computer Science Education I

Module label: Computer Science Education I

Module code: inf700

Credit points: 6.0 KP

Workload: 180 h

Used in course of study:
- Fach-Bachelor Informatik > Akzentsetzungsmodule
- Zwei-Fächer-Bachelor Informatik > Aufbaumodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

Contact person:

- Module responsibility:
  - Ira Diethelm
- Authorized examiners:
  - Ira Diethelm
  - Die im Modul Lehrenden

Entry requirements:

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.

Reader's advisory:

<table>
<thead>
<tr>
<th><strong>Links</strong></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Language of instruction</td>
<td>German</td>
<td></td>
</tr>
<tr>
<td>Duration (semesters)</td>
<td>1 Semester</td>
<td></td>
</tr>
<tr>
<td>Module frequency</td>
<td>jährlich</td>
<td></td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
<td></td>
</tr>
<tr>
<td>Modullevel</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
<td></td>
</tr>
<tr>
<td>Lern-/Lehrform / Type of program</td>
<td>V &amp; Ü</td>
<td></td>
</tr>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
<td>CS basics</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Final exam of module</strong></th>
<th>At the end of the lecture period</th>
<th>Oral exam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course type</td>
<td>Comment</td>
<td>SWS</td>
</tr>
<tr>
<td>Lecture</td>
<td></td>
<td>2.00</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>2.00</td>
</tr>
</tbody>
</table>

**Total time of attendance for the module**: 56 h
## mat950 - Discrete Mathematics

<table>
<thead>
<tr>
<th>Module label</th>
<th>Discrete Mathematics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>mat950</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
</tbody>
</table>

**Used in course of study**
- Fach-Bachelor Informatik > Aufbaumodule
- Fach-Bachelor Wirtschaftsinformatik > Aufbaumodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

### Contact person
- Module responsibility
  - Florian Heß
  - Andreas Stein
  - Sandra Stein

### Entry requirements
- 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 graph theory
- 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

### Reader's advisory
- 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
- Modulelevel: AC (Aufbaucurriculum / Composition)
- Modulart: Pflicht / Mandatory

### Lern-/Lehrform / Type of program
- Vorkenntnisse / Previous knowledge

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

Bonus points can be earned.

### Course type

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>3.00</td>
<td>WiSe</td>
<td>42 h</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>1.00</td>
<td>WiSe</td>
<td>14 h</td>
</tr>
</tbody>
</table>

**Total time of attendance for the module**: 56 h
Aufbaumodule

inf005 - Software Engineering I

<table>
<thead>
<tr>
<th>Module label</th>
<th>Software Engineering I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</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>
| Used in course of study | Fach-Bachelor Informatik > Aufbaumodule  
                        | Fach-Bachelor Mathematik > Nebenfachmodule  
                        | Fach-Bachelor Wirtschaftsinformatik > Aufbaumodule 
                        | Master of Education (Wirtschaftspädagogik) Informatik > Mastermodule 
                        | Zwei-Fächer-Bachelor Informatik > Aufbaumodule |
| Contact person        | Module responsibility                                        |
|                       | » Andreas Winter                                             |
| Authorized examiners  | » Andreas Winter                                             |
|                       | » Die im Modul Lehrenden                                     |
| Entry requirements    | The objective of the module is to convey the development and maintenance 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.

**Reader's advisory**

Helmut Balzert: Lehrbuch der Software-Technik, Spektrum Akademischer Verlag, 3. Auflage 2009

**Links**

**Language of instruction**  
German

**Duration (semesters)**  
1 Semester

**Module frequency**  
jährlich

**Module capacity**  
unlimited

**Modullevel**  
---

**Modulart**  
je nach Studiengang Pflicht oder Wahlpflicht

**Lern-/Lehrform / Type of program**  
V + Ü

**Vorkenntnisse / Previous knowledge**

<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td>At the end of the lecture period or during the lecture period (portfolio)</td>
<td>Written exam or oral exam or portfolio (3 services)</td>
</tr>
</tbody>
</table>

**Course type**  
**Comment**  
**SWS**  
**Frequency**  
**Workload attendance**

<table>
<thead>
<tr>
<th>Lecture</th>
<th>3.00</th>
<th>WiSe</th>
<th>42 h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tutorial</td>
<td>1.00</td>
<td>WiSe</td>
<td>14 h</td>
</tr>
</tbody>
</table>

**Total time of attendance for the module**  
56 h
inf010 - Computer Networks

Module label: Computer Networks

Module code: inf010

Credit points: 6.0 KP

Workload: 180 h

Used in course of study:
- Fach-Bachelor Informatik > Aufbaumodule
- Fach-Bachelor Wirtschaftsinformatik > Aufbaumodule
- Master of Education (Gymnasium) Informatik > Mastermodule
- Master of Education (Wirtschaftspädagogik) Informatik > Mastermodule
- Zwei-Fächer-Bachelor Informatik > Aufbaumodule

Contact person

Module responsibility
- Oliver Kramer

Authorized examiners
- Oliver Kramer
- Die im Modul Lehrenden

Entry requirements

Skills to be acquired in this module

Professional competence:
The students:
- Identify the layers of the ISO/OSI model
- Recognise the main concepts and algorithms of each IOS/OSI layer
- Assign technical processes to the layers
- Classify new technologies to the main concepts of the ISO/OSI model
- Compare different methods and approaches of a layer (i.e. TCP and UDP)
- Characterise safety-critical aspects of each layer

Methodological competence:
The students
- Administer small networks
- Characterise safety-critical aspects of networks

Social competence:
The students work on exercises in small teams

Self-competence:
The students recognise their administration abilities

Module contents

Contents of this lecture (cf. suggested reading Tanenbaum and Wetherall)

- Introduction to networks and the internet
- Physical Layer
- Data Link Layer
- MAC Sub-Layer
- Network Layer
- Transport Layer
- Session Layer
- Presentation Layer
- Application Layer
- Technologies (Cable and Co)
- Nyquist Shannon and Transmissions
- CDMA
- Hamming & CRC
- Stop & wait, go back n, selective repeat
- Aloha & CSMA
- Ethernet technologies
- Wifi
- Paket switchen & Dijkstra
- IP Addressing & Header
- TCP
- UDP
- Buckets & TCP-Reo
- DNS
- Flask
- RSA & PGP
- Firewalls
Reader's advisory

- lecture notes

Links
http://einstein.informatik.uni-oldenburg.de/20902.html

Language of instruction
German

Duration (semesters)
1 Semester

Module frequency
jährlich

Module capacity
unlimited

Modullevel
---

Modulart
je nach Studiengang Pflicht oder Wahlpflicht

Lern-/Lehrform / Type of program
V + Ü

Vorkenntnisse / Previous knowledge

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

Course type

<table>
<thead>
<tr>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>3.00</td>
<td>SuSe</td>
<td>42 h</td>
</tr>
<tr>
<td>Exercises</td>
<td>1.00</td>
<td>SuSe</td>
<td>14 h</td>
</tr>
</tbody>
</table>

Total time of attendance for the module
56 h
**inf700 - Computer Science Education I**

**Module label**
Computer Science Education I

**Module code**
inf700

**Credit points**
6.0 KP

**Workload**
180 h

**Used in course of study**
- Fach-Bachelor Informatik > Akzentsetzungsmodul
- Zwei-Fächer-Bachelor Informatik > Aufbaumodule
- Zwei-Fächer-Bachelor Informatik > Basismodule

**Contact person**
Module responsibility
- Ira Diethelm

**Authorized examiners**

- Ira Diethelm
- Die im Modul Lehrenden

**Entry requirements**

**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.

**Reader's advisory**
<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>jährlich</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Modullevel</td>
<td>---</td>
</tr>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
<tr>
<td>Lern-Lehrform / Type of program</td>
<td>V &amp; Ü</td>
</tr>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
<td>CS basics</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>2.00</td>
<td></td>
<td>28 h</td>
</tr>
<tr>
<td>Exercises</td>
<td></td>
<td>2.00</td>
<td></td>
<td>28 h</td>
</tr>
</tbody>
</table>

Total time of attendance for the module 56 h
inf709 - Practicals in Computer Science

Module label
Practicals in Computer Science

Module code
inf709

Credit points
12.0 KP

Workload
360 h

Used in course of study
- Zwei-Fächer-Bachelor Informatik > Aufbaumodule

Contact person
Module responsibility
- Ira Diethelm

Authorized examiners
- Lehrende der Informatik

Module counseling
- Lehrende der Informatik

Entry requirements
Skills to be acquired in this module
The objective of this module is the extension and the consolidation of computer science core concepts.

Professional competence
The students:
- analyse and/or implement computing systems or their components in a practical relevant context
- apply skilled knowledge to complex computer science tasks
- design, implement, document and present an IT-project

Methodological competence
The students:
- address the techniques of scientific work
- write and present scientific texts
- structure complex problems
- document complex problems and its solutions

Social competence
The students:
- implement solutions for complex problems in a team
- prepare and present topics in a team purposefully
- identify stages of work and are responsible for them

Self-competence
The students:
- identify and characterise tasks according to their abilities

Module contents
The module practical computer science consolidation contains of one project (9CP) and of one pro-seminar (3CP) or of one practical course (6CP), of one pro-seminar (3CP) and of one seminar (3CP). The content is related to the classes.

Reader's advisory

Links

Language of instruction
German

Duration (semesters)
1 Semester

Module frequency
halbjährlich

Module capacity
unlimited

Modullevel
AS (Akzentsetzung)

Modultyp
Wahlpflicht

Lern- / Lehreform / Type of program
1 PG, 1 S, or 1 PR, 2 S

Vorkenntnisse / Previous knowledge

Examination
Time of examination
Type of examination
<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</th>
<th>Type of examination</th>
<th>Portfolio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td>At the End of the Lecture Period</td>
<td>Portfolio</td>
<td></td>
</tr>
<tr>
<td>Course type</td>
<td>Comment</td>
<td>SWS</td>
<td>Frequency</td>
</tr>
<tr>
<td>Seminar und Projekt</td>
<td>8.00</td>
<td>SuSe and WiSe</td>
<td>112 h</td>
</tr>
<tr>
<td>Seminar und Praktikum</td>
<td>8.00</td>
<td>SuSe and WiSe</td>
<td>112 h</td>
</tr>
</tbody>
</table>

**Total time of attendance for the module** 224 h
inf800 - Proseminar in Computer Science

<table>
<thead>
<tr>
<th>Module label</th>
<th>Proseminar in Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</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>

**Used in course of study**

- Fach-Bachelor Betriebswirtschaftslehre für Leistungssportlerinnen und Leistungssportler > Fachnahe Angebote Informatik
- Fach-Bachelor Betriebswirtschaftslehre für Leistungssportler > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Betriebswirtschaftslehre mit juristischem Schwerpunkt > Fachnahe Angebote Informatik
- Fach-Bachelor Betriebswirtschaftslehre mit juristischem Schwerpunkt > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Biologie > Fachnahe Angebote Informatik
- Fach-Bachelor Biologie > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Business Administration in mittelständischen Unternehmen > Fachnahe Angebote Informatik
- Fach-Bachelor Business Administration in mittelständischen Unternehmen > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Chemie > Fachnahe Angebote Informatik
- Fach-Bachelor Chemie > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Comparative and European Law > Fachnahe Angebote Informatik
- Fach-Bachelor Comparative and European Law > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Engineering Physics > Fachnahe Angebote Informatik
- Fach-Bachelor Engineering Physics > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Informatik > Fachnahe Angebote Informatik
- Fach-Bachelor Informatik > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Interkulturelle Bildung und Beratung > Fachnahe Angebote Informatik
- Fach-Bachelor Interkulturelle Bildung und Beratung > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Mathematik > Fachnahe Angebote Informatik
- Fach-Bachelor Mathematik > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Nachhaltigkeitsökonomik > Fachnahe Angebote Informatik
- Fach-Bachelor Nachhaltigkeitsökonomik > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Pädagogik > Fachnahe Angebote Informatik
- Fach-Bachelor Pädagogik > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft > Fachnahe Angebote Informatik
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Physik > Fachnahe Angebote Informatik
- Fach-Bachelor Physik > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Physik, Technik und Medizin > Fachnahe Angebote Informatik
- Fach-Bachelor Physik, Technik und Medizin > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Sozialwissenschaften > Fachnahe Angebote Informatik
- Fach-Bachelor Sozialwissenschaften > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Umweltwissenschaften > Fachnahe Angebote Informatik
- Fach-Bachelor Umweltwissenschaften > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Wirtschaftsinformatik > Fachnahe Angebote Informatik
- Fach-Bachelor Wirtschaftsinformatik > Fachnahe Angebote Wirtschaftsinformatik
- Fach-Bachelor Wirtschaftswissenschaften > Fachnahe Angebote Informatik
- Fach-Bachelor Wirtschaftswissenschaften > Fachnahe Angebote Wirtschaftsinformatik
- Master of Education (Wirtschaftspädagogik) Informatik > Praktische Vertiefung der Informatik
- Zwei-Fächer-Bachelor Anglistik > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Anglistik > Fachnahe Angebote Wirtschaftsinformatik
- Zwei-Fächer-Bachelor Biologie > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Biologie > Fachnahe Angebote Wirtschaftsinformatik
- Zwei-Fächer-Bachelor Chemie > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Chemie > Fachnahe Angebote Wirtschaftsinformatik
- Zwei-Fächer-Bachelor Elementarmathematik > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Elementarmathematik > Fachnahe Angebote Wirtschaftsinformatik
- Zwei-Fächer-Bachelor Ev. Theologie und Religionspädagogik > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Ev. Theologie und Religionspädagogik > Fachnahe Angebote Wirtschaftsinformatik
- Zwei-Fächer-Bachelor Gender Studies > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Gender Studies > Fachnahe Angebote Wirtschaftsinformatik
- Zwei-Fächer-Bachelor Germanistik > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Germanistik > Fachnahe Angebote Wirtschaftsinformatik
- Zwei-Fächer-Bachelor Geschichte > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Geschichte > Fachnahe Angebote Wirtschaftsinformatik
- Zwei-Fächer-Bachelor Informatik > Aufbaumodule
- Zwei-Fächer-Bachelor Informatik > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Informatik > Fachnahe Angebote Wirtschaftsinformatik
- Zwei-Fächer-Bachelor Interdisziplinäre Sachbildung > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Interdisziplinäre Sachbildung > Fachnahe Angebote Wirtschaftsinformatik
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

<table>
<thead>
<tr>
<th>Module contents</th>
<th>according to the assigned task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reader's advisory</td>
<td>according to the assigned task</td>
</tr>
</tbody>
</table>

<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>
<tr>
<td>Modullevel</td>
</tr>
<tr>
<td>Modulart</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vorkenntnisse / Previous knowledge</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td></td>
<td>Presentation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seminar</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>SuSe and WiSe</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>28 h</td>
</tr>
</tbody>
</table>
Praktische Vertiefung

inf004 - Software Project

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

Used in course of study

- Fach-Bachelor Betriebswirtschaftslehre für Leistungssportlerinnen und Leistungssportler > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Betriebswirtschaftslehre mit juristischem Schwerpunkt > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Biologie > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Business Administration in mittelständischen Unternehmen > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Chemie > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Comparative and European Law > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Engineering Physics > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Informatik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Interkulturelle Bildung und Beratung > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Mathematik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Nachhaltigkeitsökonomik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Pädagogik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Physik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Physik, Technik und Medizin > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Pädagogik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Interkulturelle Bildung und Beratung > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Mathematik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Wirtschaftsinformatik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Wirtschaftswissenschaften > Praxismodule für Studierende mit außerschulchem Berufsziel
- Fach-Bachelor Wirtschaftswissenschaften > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Anglistik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Biologie > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Chemie > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Elementarmathematik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Ev. Theologie und Religionspädagogik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Gender Studies > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Germanistik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Geschichts > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Informatik > Praktische Vertiefung
- Zwei-Fächer-Bachelor Interdisziplinäre Sachbildung > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Kunst und Medien > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Materielle Kultur: Textil > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Mathematik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Niederländistik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Ökonomische Bildung > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Pädagogik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Philosophie / Werte u. Normen > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Politik-Wirtschaft > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Slavistik > Praxismodule für Studierende mit außerschulchem Berufsziel
- Zwei-Fächer-Bachelor Sonderpädagogik > Praxismodule für Studierende mit außerschulchem Berufsziel
Contact person

Module responsibility

» Marco Grawunder

Authorized examiners

» Die im Modul Lehrenden

Module counseling

» Die im Modul Lehrenden

Entry requirements

Java Programmierkurs, Software Engineering, Algorithmen und Datenstrukturen

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

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.

Reader's advisory

Links

https://confluence.swp.offis.uni-oldenburg.de/display/SWPWP

Language of instruction

German

Duration (semesters)

2 Semester

Module frequency

jährlich

Module capacity

unlimited

Modullevel

PB (Professionalisierungsbereich)

Modulart

Ergänzung/Professionalisierung

Lern-/Lehrform / Type of program

V + Ü + P
<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>2.00</td>
<td></td>
<td>SuSe</td>
<td>28 h</td>
</tr>
<tr>
<td>Exercises</td>
<td>2.00</td>
<td></td>
<td>SuSe and WiSe</td>
<td>28 h</td>
</tr>
<tr>
<td>Project</td>
<td>4.00</td>
<td></td>
<td>SuSe and WiSe</td>
<td>56 h</td>
</tr>
</tbody>
</table>

**Total time of attendance for the module**

112 h
inf009 - Database Practical

<table>
<thead>
<tr>
<th>Module label</th>
<th>Database Practical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</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>
<tr>
<td>Used in course of study</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fach-Bachelor Informatik &gt; Akzentsetzungsmodulle</td>
</tr>
<tr>
<td></td>
<td>Fach-Bachelor Wirtschaftsinformatik &gt; Akzentsetzungsmodulle</td>
</tr>
<tr>
<td></td>
<td>Master of Education (Wirtschaftspädagogik) Informatik &gt; Praktische Vertiefung der Informatik</td>
</tr>
<tr>
<td></td>
<td>Zwei-Fächer-Bachelor Informatik &gt; Praktische Vertiefung</td>
</tr>
<tr>
<td>Contact person</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Module responsibility</td>
</tr>
<tr>
<td></td>
<td>▪ Marco Grawunder</td>
</tr>
<tr>
<td></td>
<td>Authorized examiners</td>
</tr>
<tr>
<td></td>
<td>▪ Marco Grawunder</td>
</tr>
<tr>
<td></td>
<td>▪ Die im Modul Lehrenden</td>
</tr>
<tr>
<td>Entry requirements</td>
<td></td>
</tr>
<tr>
<td>Skills to be acquired in this module</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Professional competence</td>
</tr>
<tr>
<td></td>
<td>The students:</td>
</tr>
<tr>
<td></td>
<td>▪ Realise, implement and program data base systems</td>
</tr>
<tr>
<td></td>
<td>▪ Program and implement database-oriented system routines</td>
</tr>
<tr>
<td></td>
<td>▪ Implement optimisation goals in the modelling phase</td>
</tr>
<tr>
<td></td>
<td>▪ Administer professional database systems (installation, maintenance and adjustment)</td>
</tr>
<tr>
<td></td>
<td>▪ Recognise database systems' performance problems and are able to fix them with according methods</td>
</tr>
<tr>
<td></td>
<td>▪ Organise and control processes of database systems</td>
</tr>
<tr>
<td></td>
<td>Social competence</td>
</tr>
<tr>
<td></td>
<td>The students:</td>
</tr>
<tr>
<td></td>
<td>▪ Solve database system problems in a team</td>
</tr>
<tr>
<td></td>
<td>Self-competence</td>
</tr>
<tr>
<td></td>
<td>The students:</td>
</tr>
<tr>
<td></td>
<td>▪ Acknowledge the limits of their ability to cope with pressure during the implementation and are aware of failures</td>
</tr>
<tr>
<td></td>
<td>▪ Reflect their self-perception</td>
</tr>
<tr>
<td>Module contents</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Contents of this module are:</td>
</tr>
<tr>
<td></td>
<td>▪ System-oriented database management programming,</td>
</tr>
<tr>
<td></td>
<td>▪ Implementation of catalogue systems,</td>
</tr>
<tr>
<td></td>
<td>▪ Optimisation strategies based on parallelisation and partitioning requirements</td>
</tr>
<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>Modul frequency</td>
<td>jährlich</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Modullevel</td>
<td>---</td>
</tr>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
<tr>
<td>Lern-/Lehrform / Type of program</td>
<td>P</td>
</tr>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
<td></td>
</tr>
<tr>
<td>Examination</td>
<td>Time of examination</td>
</tr>
<tr>
<td>Final exam of module</td>
<td>At the end of the lecture period</td>
</tr>
<tr>
<td>Course type</td>
<td>Practical</td>
</tr>
<tr>
<td>SWS</td>
<td>4.00</td>
</tr>
<tr>
<td>Frequency</td>
<td>WiSe</td>
</tr>
<tr>
<td>Workload attendance</td>
<td>56 h</td>
</tr>
</tbody>
</table>
inf014 - Operating Systems Practical

<table>
<thead>
<tr>
<th>Module label</th>
<th>Operating Systems Practical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</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>
</tbody>
</table>

**Used in course of study**
- Fach-Bachelor Informatik > Akzentsetzungsmodul
- Fach-Bachelor Wirtschaftsinformatik > Akzentsetzungsmodul
- Master of Education (Wirtschaftspädagogik) Informatik > Praktische Vertiefung der Informatik
- Zwei-Fächer-Bachelor Informatik > Praktische Vertiefung

**Contact person**

- Module responsibility
  - Oliver Theel
- Authorized examiners
  - Oliver Theel
  - Die im Modul Lehrenden

**Entry requirements**

**Skills to be acquired in this module**
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.

**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 own shortcomings in their initial ability to successfully transfer theory to praxis

**Module contents**
The contents of this module are:

- Analysis of a rudimentary operating system
- Design and implementation of a process management subsystem
- Design and implementation of process synchronisation mechanisms
- Design and implementation of a virtual memory management subsystem
- Design and implementation of a file subsystem or dialog subsystem
<table>
<thead>
<tr>
<th></th>
<th></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>jährlich</td>
</tr>
<tr>
<td>Module capacity</td>
<td>unlimited</td>
</tr>
<tr>
<td>Reference text</td>
<td><strong>Associated with the modules:</strong></td>
</tr>
<tr>
<td></td>
<td>- Betriebssysteme I</td>
</tr>
<tr>
<td></td>
<td>- Betriebssysteme II</td>
</tr>
<tr>
<td></td>
<td>- Verteilte Systeme</td>
</tr>
<tr>
<td>Modulelevel</td>
<td>AS (Akzentsetzung / Accentuation)</td>
</tr>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
<tr>
<td>Lern-/Lehrform / Type of program</td>
<td>P</td>
</tr>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
<td>- Betriebssysteme I</td>
</tr>
<tr>
<td></td>
<td>- Betriebssysteme II</td>
</tr>
<tr>
<td></td>
<td>- Computer language: C, Assembler</td>
</tr>
<tr>
<td>Examination</td>
<td></td>
</tr>
<tr>
<td>Time of examination</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 semester</td>
</tr>
<tr>
<td>Course type</td>
<td>Practical</td>
</tr>
<tr>
<td>SWS</td>
<td>4.00</td>
</tr>
<tr>
<td>Frequency</td>
<td>WiSe</td>
</tr>
<tr>
<td>Workload attendance</td>
<td>56 h</td>
</tr>
</tbody>
</table>
# inf018 - Media Processing

<table>
<thead>
<tr>
<th>Module label</th>
<th>Media Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module code</td>
<td>inf018</td>
</tr>
<tr>
<td>Credit points</td>
<td>6.0 KP</td>
</tr>
<tr>
<td>Workload</td>
<td>180 h</td>
</tr>
</tbody>
</table>

## Used in course of study

- Fach-Bachelor Betriebswirtschaftslehre für Leistungssportlerinnen und Leistungssportler > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Betriebswirtschaftslehre mit juristischem Schwerpunkt > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Biologie > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Business Administration in mittelständischen Unternehmen > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Chemie > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Comparative and European Law > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Engineering Physics > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Informatik > Akzentsetzungsmodule
- Fach-Bachelor Informatik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Interkulturelle Bildung und Beratung > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Mathematik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Nachhaltigkeitsökonomik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Pädagogik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Physik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Physik, Technik und Medizin > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Sozialwissenschaften > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Umweltwissenschaften > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Wirtschaftswissenschaften > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Wirtschaftswissenschaften > Akzentsetzungsmodule
- Fach-Bachelor Wirtschaftswissenschaften > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Master of Education (Gymnasium) Informatik > Mastermodule
- Master of Education (Wirtschaftspädagogik) Informatik > Mastermodule
- Master of Education (Wirtschaftspädagogik) Informatik > Praktische Vertiefung der Informatik
- Master Wirtschaftsinformatik > Bereichswahlmodule
- Zwei-Fächer-Bachelor Anglistik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Biologie > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Chemie > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Elementarmathematik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Ev. Theologie und Religionspädagogik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Germanistik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Geschichte > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Informatik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Informatik > Praktische Vertiefung
- Zwei-Fächer-Bachelor Interdisziplinäre Sachbildung > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Kunst und Medien > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Materielle Kultur: Textil > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Mathematik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Musik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Niederlandistik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Ökonomische Bildung > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Pädagogik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Philosophie / Werte u. Normen > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Physik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Politik-Wirtschaft > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Slavistik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Sonderpädagogik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Sozialwissenschaften > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Sportwissenschaft > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Technik > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
Zwei-Fächer-Bachelor Wirtschaftswissenschaften > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"

Contact person
Module responsibility
- Susanne Boll-Westermann

Authorized examiners
- Susanne Boll-Westermann
- Die im Modul Lehrenden

Entry requirements
Skills to be acquired in this module
Professional competence:
The students:
- name the basic concepts and characteristics of digital media
- name the core concepts of encoding and compressing images, videos and audio files
- characterise the complexity of the analysis, classification and processing of unstructured media, using the examples of image analysis
- apply concepts of encoding, compression and image analysis independently

Module contents
Media processing technologies are presented in the lecture. One focus of the lecture is the encoding of digital images and the compression of an image, image enhancement and image processing. The lecture also deals with encoding and analysis of video and audio. This lecture is accompanied by simple practical tasks.

Reader's advisory
- Reserve shelf in the library; extensive list of links in e-learning platform StudIP covering course topics.

Links
https://www.uni-oldenburg.de/informatik/medieninformatik/lehre/

Language of instruction
German

Duration (semesters)
1 Semester

Module frequency
jährlich

Module capacity
unlimited

Modullevel
AS (Akzentsetzung / Accentuation)

Modulart
je nach Studiengang Pflicht oder Wahlpflicht

Lern-/Lehrform / Type of program
V+P

Vorkenntnisse / Previous knowledge

<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam of module</td>
<td>At the end of the lecture period</td>
<td>Project and oral exam</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course type</th>
<th>Comment</th>
<th>SWS</th>
<th>Frequency</th>
<th>Workload attendance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td></td>
<td>2.00</td>
<td>WiSe</td>
<td>28 h</td>
</tr>
<tr>
<td>Project</td>
<td></td>
<td>2.00</td>
<td>WiSe</td>
<td>28 h</td>
</tr>
<tr>
<td>Course type</td>
<td>Comment</td>
<td>SWS</td>
<td>Frequency</td>
<td>Workload attendance</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>---------</td>
<td>-----</td>
<td>-----------</td>
<td>---------------------</td>
</tr>
<tr>
<td><strong>Total time of attendance for the module</strong></td>
<td></td>
<td></td>
<td></td>
<td>56 h</td>
</tr>
</tbody>
</table>
inf021 - Advanced Java Technology Practical

Module label  Advanced Java Technology Practical
Module code  inf021
Credit points  6.0 KP
Workload  180 h

Used in course of study
- Fach-Bachelor Informatik > Akzentsetzungsmodule
- Fach-Bachelor Wirtschaftsinformatik > Akzentsetzungsmodule
- Master of Education (Wirtschaftspädagogik) Informatik > Praktische Vertiefung der Informatik
- Zwei-Fächer-Bachelor Informatik > Praktische Vertiefung

Contact person
- Module responsibility
  - Dietrich Boles
- Authorized examiners
  - Dietrich Boles
  - Die im Modul Lehrenden

Entry requirements

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

Professional competence:
The students:
- Name the essential packages of the JDK class library
- Structure large-scale 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-scale programs
- Evaluate the quality of large-scale programs related to their maintainability, reusability and expandability

Methodological competence:
The students:
- Search for solutions on the internet

Social competence:
The students:
- Discuss own and someone else's solutions

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 practical course:

- GUI (AWT, Swing, JavaFX)
- Java-Basics and Collection-API
- Graphics and multimedia
- Events
- Model-View-Control (MVC)
- Threads
- Internationalisation, localization
- Reflection
- IO, Files
- Tools (compiler, classloader, printer, ...)
- Storage technologies (XML and serialisation)
- Distributed programming (sockets and RMI)
- Databases (JDBC)
- Compression
- Security concepts
The practical course is based on a large-scale project. This project is developed step-by-step relating to the subjects of the course.

**Reader's advisory**

**Links**  
http://www.boles.de/teaching/javapraktikum/index.html

**Language of instruction**  
German

**Duration (semesters)**  
1 Semester

**Module frequency**  
jährlich

**Module capacity**  
unlimited

**Module level**  
---

**Modulart**  
je nach Studiengang Pflicht oder Wahlpflicht

**Lern-/Lehrform / Type of program**  
P

**Vorkenntnisse / Previous knowledge**  
Programmkurs

**Examination**  
Time of examination  
Type of examination

<table>
<thead>
<tr>
<th>Final exam of module</th>
<th>At the end of the lecture period</th>
<th>Hands-on training</th>
</tr>
</thead>
</table>

**Course type**  
Practical

**SWS**  
4.00

**Frequency**  
WiSe

**Workload attendance**  
56 h
inf202 - Computer Engineering Practical

Module label: Computer Engineering Practical
Module code: inf202
Credit points: 6.0 KP
Workload: 180 h

Used in course of study:
- Master of Education (Gymnasium) Informatik > Mastermodule
- Master of Education (Wirtschaftspädagogik) Informatik > Praktische Vertiefung der Informatik
- Zwei-Fächer-Bachelor Informatik > Praktische Vertiefung

Contact person:
- Module responsibility
  - Alfred Mikschl
  - Die im Modul Lehrenden
- Authorized examiners
  - Alfred Mikschl
  - Die im Modul Lehrenden

Entry requirements

Skills to be acquired in this module

Module contents

Reader's advisory

Links

Language of instruction: German
Duration (semesters): 1 Semester

Module frequency

Module capacity: unlimited

Modulart: je nach Studiengang Pflicht oder Wahlpflicht

Lehr-/Lehrform / Type of program

Vorkenntnisse / Previous knowledge

Examination

<table>
<thead>
<tr>
<th>Final exam of module</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PK</td>
</tr>
</tbody>
</table>

Course type: Practical

SWS: 4.00
Frequency: SuSe
Workload attendance: 56 h
inf406 - Laboratory Real-Time Systems

Module label Laboratory Real-Time Systems
Module code inf406
Credit points 6.0 KP
Workload 180 h
Used in course of study
- Fach-Bachelor Informatik > Akzentsetzungsmodule
- Master of Education (Wirtschaftspädagogik) Informatik > Praktische Vertiefung der Informatik
- Zwei-Fächer-Bachelor Informatik > Praktische Vertiefung

Contact person
Module responsibility
- Ernst-Rüdiger Olderog
Authorized examiners
- Ernst-Rüdiger Olderog
- Die im Modul Lehrenden

Entry requirements
Theoretische Informatik I und II

Skills to be acquired in this module
The students learn about methods and tools, and how to apply, specify, simulate, verify, and implement real-time systems (RTS). The students gain hands-on experience using tangible Mini-Robots (Lego Mindstorms).

Professional competence
The students:
- implement RTS with Lego Mindstorm Robots NXT
- simulate and verify RTS on the basis of real-time automata with the model checker UPPAAL
- apply the tool Moby/RT to specify and simulate RTS on the basis of PLC-Automata, and to translate them into Java-Code for Lego Mindstorms NXT and into UPPAAL

Methodological competence
The students:
- realise control tasks with Lego Mindstorms
- specify RTS as networks of real-time automata and verify them with UPPAAL
- design RTS using Moby/RT
- realise systematically sophisticated time-dependent control tasks with Moby/RT, Lego Mindstorms, and UPPAAL

Social competence
The students:
- solve tasks in a team
- present solutions and discuss them

Self-competence
The students:
- recognise (sub-)problems of RTS and are responsible for their realisation

Module contents
Real-time-systems are systems, where the time at which an output is generated or at which data are read is of importance. Compared to usual programming methods, RTS models are extended by the additional dimension of time. An example for a RTS is an airbag in a car, which needs to be triggered at the right moment of time, not too early and not too late, because the effect of the airbag is useful only for a few hundredths of seconds.

The course introduces methods and tools which are then practically applied to specify, verify, and implement RTS. The students gain hands-on experience using Mini-Robots (Lego-Mindstorms) to implement RTS.

Reader's advisory

Links

Language of instruction German
Duration (semesters) 1 Semester
Module frequency unregelmäßig
Module capacity unlimited
<table>
<thead>
<tr>
<th>Modullevel</th>
<th>---</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulart</td>
<td>je nach Studiengang Pflicht oder Wahlpflicht</td>
</tr>
<tr>
<td>Lern-/Lehrform / Type of program</td>
<td>P</td>
</tr>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
<td></td>
</tr>
<tr>
<td>Examination</td>
<td>Time of examination</td>
</tr>
<tr>
<td>Final exam of module</td>
<td>At the end of the lecture period</td>
</tr>
<tr>
<td>Course type</td>
<td>Practical</td>
</tr>
<tr>
<td>SWS</td>
<td>4.00</td>
</tr>
<tr>
<td>Frequency</td>
<td>WiSe</td>
</tr>
<tr>
<td>Workload attendance</td>
<td>56 h</td>
</tr>
</tbody>
</table>
inf803 - Special Topics in Computer Science I

**Module label**
Special Topics in Computer Science I

**Module code**
inf803

**Credit points**
6.0 KP

**Workload**
180 h

**Used in course of study**
- Fach-Bachelor Informatik > Akzentsetzungsmodule
- Master of Education (Wirtschaftspädagogik) Informatik > Praktische Vertiefung der Informatik
- Zwei-Fächer-Bachelor Informatik > Praktische Vertiefung

**Contact person**
Module responsibility
- Die im Modul Lehrenden

**Authorized examiners**
- Die im Modul Lehrenden

**Entry 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

**Module contents**
According to the assigned task

**Reader's advisory**
According to the assigned task

**Languages of instruction**
German, English

**Duration (semesters)**
1 Semester

**Module frequency**
halbjährlich

**Module capacity**
unlimited

**Modulart**
je nach Studiengang Pflicht oder Wahlpflicht

**Lern-/Lehrform / Type of program**
2 courses out of VL, Ü, Tut, SE, PR

**Vorkenntnisse / Previous knowledge**

**Examination**
Time of examination
Type of examination
Final exam of module
Exercises or presentation or oral exam or written exam

**Course type**
Course selection
<table>
<thead>
<tr>
<th>SWS</th>
<th>4.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>SuSe or WiSe</td>
</tr>
<tr>
<td>Workload attendance</td>
<td>56 h</td>
</tr>
</tbody>
</table>
Inf804 - Special Topics in Computer Science II

Module label: Special Topics in Computer Science II
Module code: inf804
Credit points: 6.0 KP
Workload: 180 h

Used in course of study:
- Fach-Bachelor Informatik > Akzentsetzungsmoduls
- Master of Education (Wirtschaftspädagogik) Informatik > Praktische Vertiefung der Informatik
- Zwei-Fächer-Bachelor Informatik > Praktische Vertiefung

Contact person:
- Module responsibility: Die im Modul Lehrenden
- Authorized examiners: Die im Modul Lehrenden

Entry 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

Module contents:
According to the assigned task

Reader’s advisory:
According to the assigned task

Languages of instruction:
German, English

Duration (semesters):
1 Semester

Module frequency:
halbjährlich

Module capacity:
unlimited

Module level:
---

Modulart:
je nach Studiengang Pflicht oder Wahlpflicht

Lern-/Lehrform / Type of program:
2 courses out of VL, Ü, Tut, SE, PR

Vorkenntnisse / Previous knowledge:

Examination:
Type of examination:
Final exam of module:
Exercises or presentation or oral exam or written exam

Course type:
Course selection
<table>
<thead>
<tr>
<th>SWS</th>
<th>4.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>SuSe or WiSe</td>
</tr>
<tr>
<td>Workload attendance</td>
<td>56 h</td>
</tr>
</tbody>
</table>
inf808 - Current Topics in Computer Science

Module label: Current Topics in Computer Science
Module code: inf808
Credit points: 3.0 KP
Workload: 90 h

Used in course of study:
- Fach-Bachelor Informatik > Akzentsetzungsmodul
- Master of Education (Wirtschaftspädagogik) Informatik > Praktische Vertiefung der Informatik
- Zwei-Fächer-Bachelor Informatik > Praktische Vertiefung

Contact person:
Module responsibility:
- Die im Modul Lehrenden

Authorized examiners:
- Die im Modul Lehrenden

Entry 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 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
Reader’s advisory:
According to the assigned task

Links:

Language of instruction: German
Duration (semesters): 1 Semester
Module frequency: unregelmäßig
Module capacity: unlimited
Modullevel: ---

Lern-/Lehrform / Type of program:
- je nach Studiengang Pflicht oder Wahlpflicht

Vorkenntnisse / Previous knowledge:
1 course out of VL, SE, PR

Examination:
Final exam of module
Time of examination:
Type of examination:
Exercises or presentation or oral exam or written
<table>
<thead>
<tr>
<th>Examination</th>
<th>Time of examination</th>
<th>Type of examination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>exam</td>
</tr>
<tr>
<td><strong>Course type</strong></td>
<td>Course selection</td>
<td></td>
</tr>
<tr>
<td><strong>SWS</strong></td>
<td>2.00</td>
<td></td>
</tr>
<tr>
<td><strong>Frequency</strong></td>
<td>SuSe and WiSe</td>
<td></td>
</tr>
<tr>
<td><strong>Workload attendance</strong></td>
<td>28 h</td>
<td></td>
</tr>
</tbody>
</table>
Abschlussmodul
bam - Bachelor Thesis and Colloquium

Module label: Bachelor Thesis and Colloquium
Module code: bam
Credit points: 15.0 KP
Workload: 450 h

Used in course of study:
- Zwei-Fächer-Bachelor Informatik > Abschlussmodul

Contact person:
Module responsibility
- Ira Diethelm
- Die im Modul Lehrenden

Authorized examiners
- Die im Modul Lehrenden
- Die Modulverantwortlichen

Module counseling
- Die im Modul Lehrenden

Entry 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.

Reader's advisory:
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: halbjährlich
Module capacity: unlimited
Modul level: Abschlussmodul (Abschlussmodul)
<table>
<thead>
<tr>
<th>Modulart</th>
<th>Pflicht</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lern-/Lehrform / Type of program</td>
<td></td>
</tr>
<tr>
<td>Vorkenntnisse / Previous knowledge</td>
<td></td>
</tr>
<tr>
<td>Examination</td>
<td>Time of examination</td>
</tr>
<tr>
<td>Final exam of module</td>
<td>varying</td>
</tr>
<tr>
<td>Course type</td>
<td>Seminar</td>
</tr>
<tr>
<td>SWS</td>
<td>2.00</td>
</tr>
<tr>
<td>Frequency</td>
<td>28 h</td>
</tr>
<tr>
<td>Workload attendance</td>
<td>28 h</td>
</tr>
</tbody>
</table>