Modulhandbuch

Computing Science - Dual-Subject Bachelor's Programme

im Wintersemester 2022/2023

erstellt am 07/02/23

Basismodule

inf030 - Programming, Algorithms and Data Structures

Module label	Programming, Algorithms and Data Structures
Modulkürzel	inf030
Credit points	9.0 KP
Workload	270 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Basiscurriculum • Bachelor's Programme Computing Science (Bachelor) > Basismodule • Bachelor's Programme Economics and Business Administration (Bachelor) > Studienrichtung Wirtschaftsinformatik • Bachelor's Programme Mathematics (Bachelor) > Nebenfachmodule • Bachelor's Programme Sustainability Economics (Bachelor) > Wahlpflichtbereich • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule
Zuständige Personen	Lehrenden, Die im Modul (Prüfungsberechtigt) Schönberg, Christian (Module responsibility)
Prerequisites	
Skills to be acquired in this module	

Programming is one of the basic activities of computer scientists and a prerequisite for many other courses in computer science studies. The aim of the module "Programming, Data Structures and Algorithms" is to learn the basic concepts of imperative, procedural and object-oriented programming using the Java programming language and to present known, efficient algorithms and data structures for various, frequently occurring problems.

After completing the module, the students should be able to independently develop imperative and simple object-oriented programmes based on Java for solving smaller problems and assess the efficiency of their programmes. They should also be able to apply important algorithms and select them based on their complexity.

Professional competence
The students:

- describe basic concepts of imperative programming with Java

- recognise imperative programming terminology and use the appropriate terms accurately in discussions

- recognise basic terminology of object-oriented programming

- describe what programs presented to them do

- independently develop programs to solve small problems

- systematically examine their own and other people's programmes for errors

- use modern programme development environments to develop and test programmes

- create algorithms with general design concepts (e.g. Greedy method, divide-and-conquer method)

- name algorithms and data structures for solving common problems and evaluate their applicability

- name problems of efficiency of algorithmic solutions of concrete problems and evaluate them

- make a well-founded choice of an algorithm and a data structure for solving a concrete problem

- apply the learned algorithms and data structures sensibly to given and concrete problems

Methodological competence
The students:

- solve given problems from the point of view of imperative or object-oriented programming

- transfer practical experience in programme development to new tasks

Social competence
The students:

- communicate the structure and mode of operation of self-developed programmes to others

- present solutions to small tasks in front of groups

Self-competence
The students:

- organise themselves in finding algorithmic solutions to small and medium-sized problems in computer science

- incorporate the concepts of general programme design in their actions

Module contents

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

- Algorithm, programming languages, computer

- development tools, development phases

- compilers

- grammars

- logic

The second part deals with basic programming concepts:

- data types

- variables

- Expressions, statements

- control structures

- Methods, parameters

- recursion

- Reference data types, arrays

- Classes, objects

- Documentation

- Testing

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

- Methods for searching for keys, as well as insertion and deletion in dynamic data sets, e.g. lists, trees, AVL trees or hash methods,

- Methods for searching for text patterns,

- Methods for sorting data by key values, e.g. QuickSort and HeapSort,

- Graph-based applications, e.g. for finding shortest paths in graphs.

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

Literaturempfehlungen

Essential:

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

Good secondary literature:

- Dietmar Ratz, Jens Scheffler, Detlev Seese, Jan Wiesenberger: Grundkurs Programmieren in Java, Carl Hanser Verlag.
- Joachim Goll, Cornelia Heinisch: Java als erste Programmiersprache, Springer Vieweg Verlag
- Ottmann, Widmayer: Algorithmen und Datenstrukturen. Spektrum Verlag, 5. Auflage, 2012
- Segdewick, Wayne: Algorithms. Addison Wesley, 4th ed., 2011
- Siege: Einführung in die Informatik. Shaker Verlag, 2013

Links

Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	WiSe
Module capacity	unlimited
Modullevel / module level	BC (Basiscurriculum / Base curriculum)
Modulart / typ of module	Pflicht / Mandatory
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the End of the Semester	Portfolio / Klausur /mündliche Prüfung

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		4	WiSe	56
Exercises		2	WiSe	28
Präsenzzeit Modul insgesamt				84 h

inf031 - Object-oriented Modelling and Programming

Module label	Object-oriented Modelling and Programming
Modulkürzel	inf031
Credit points	9.0 KP
Workload	270 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Basiscurriculum • Bachelor's Programme Computing Science (Bachelor) > Basismodule • Bachelor's Programme Economics and Business Administration (Bachelor) > Studienrichtung Wirtschaftsinformatik • Bachelor's Programme Mathematics (Bachelor) > Nebenfachmodule • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule
Zuständige Personen	Schönberg, Christian (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
Skills to be acquired in this module	

Object orientation represents the state of the art in software development today. Given problems are first transformed into an object-oriented model and then into an object-oriented program with the help of object-oriented analysis and design methods. Aim of the module "Object-Oriented Modelling and Programming" is to learn basic concepts of object-oriented modelling using UML as the modelling notation and object-oriented programming using the Java programming language. After completing the module, students should be able to independently develop object-oriented programs based on Java for solving medium-sized problems.

Professional competence:
The students:
- know basic concepts of object-oriented modelling and UML as modelling notation

- know basic concepts of object-oriented programming with Java

- know the terminology of object-oriented modelling and programming and use the appropriate terms precisely in discussions

- can describe what object-oriented programmes presented to them do

- independently develop models and programmes for solving medium-sized problems

- systematically examine their own and other people's models and programmes for errors

- use modern development environments for modelling and developing programmes

- know the differences between the imperative, object-oriented, functional, logical and rule-based programming paradigms

Methodological competence:
The students:

- independently develop programmes for given problems by consistently applying the concepts of object-oriented modelling and programming

- transfer practical experience in programme development to new tasks

- independently develop programmes with concurrency

- can independently apply known solution methods to complex problems

Social competence:
The students:
-communicate the structure and mode of action of self-developed models and programmes to others
- present independently developed solutions to groups

Self-competence:
The students:
- organise themselves when developing programmes for small and medium-sized problems in computer science
- incorporate the concepts of object-oriented programme design in their actions

Module contents

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

- Models and modelling

- UML class diagrams

- Classes and objects

- data encapsulation

- inheritance

- Polymorphism and dynamic binding

- Exception handling

- Genericity

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

- Java Collection API

- IO and Streams

- GUI applications with JavaFX

- Parallel programming with threads

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

- Backtracking, Branch and Bound, Greedy

- Local Search, Evolutionary Algorithms

- Functional programming (e.g. Java-Lamdas, Standard ML)

- Logical programming (e.g. Prolog)

- Rule-based programming (e.g. Drools)

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

Literaturempfehlungen

Essential:

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

Good secondary literature:

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

Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	SoSe
Module capacity	unlimited

Modullevel / module level					
Modulart / typ of module					
Lehr-/Lernform / Teaching/Learning method	V+Ü				
Vorkenntnisse / Previous knowledge					
Examination		Prüfungszeiten		Type of examination	
Final exam of module		At the end of the Semester.		Portfolio / Klausur / mündl. Prüfung	
Form of instruction	Comment	SWS		Frequency	Workload of compulsory attendance
Lecture		4		SoSe	56
Exercises		2		SoSe	28
Präsenzzeit Modul insgesamt					84 h

inf200 - Computer Engineering I

Module label	Computer Engineering I
Modulkürzel	inf200
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Basismodule • Bachelor's Programme Mathematics (Bachelor) > Nebenfachmodule • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
Zuständige Personen	Nebel, Wolfgang (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
Skills to be acquired in this module	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 analyse 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 t 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.
Literaturempfehlungen	- handout manuscript of the course - Schiffmann, W.; Schmitz, R. (2001): Technische Informatik I, II, Übungsbuch; Springer Verlag, Berlin. - Dal Cin, M. (1996): Rechnerarchitektur; B.G. Teubner. - Lagemann, K. (1987): Rechnerstrukturen; Springer-Verlag, Berlin. - Oberschelp, W.; Vossen, G. (1989): Rechneraufbau und Rechnerstrukturen; Oldenbourg-Verlag. - Mano, Morris M.(1993): Computer System Architecture 3; Prentice Hall. - Gajski, D.(1997): Principles of Digital Design; Prentice Hall. - Patterson, D.A.; Hennessy, J.L. (1997): Computer Organization and Design: - The Hardware/Software Interface; 2. Edition; Morgan Kaufmann Publishers. - Wilkinson, B. (1996): Computer Architecture Design and Performance; 2. Edition; Prentice Hall. - Tannenbaum, A.S.(1999): Structured Computer Organization; 4. Edition; Prentice Hall.
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited

Modullevel / module level		BC (Basiscurriculum / Base curriculum)			
Modulart / typ of module					
Lehr-/Lernform / Teaching/Learning method		V+Ü			
Vorkenntnisse / Previous knowledge					
Examination		Prüfungszeiten		Type of examination	
Final exam of module		At the end of the semester		Written or oral exam	
Form of instruction	Comment	SWS		Frequency	Workload of compulsory attendance
Lecture		3		WiSe	42
Exercises		1		WiSe	14
Präsenzzeit Modul insgesamt					56 h

inf700 - Computer Science Education I

Module label	Computer Science Education I
Modulkürzel	inf700
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Wahlbereich Informatik, Kultur und Gesellschaft • Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Bachelor's Programme Computing Science (Bachelor) > Wahlbereich Informatik, Kultur und Gesellschaft • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule (60 KP) • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule
Zuständige Personen	Diethelm, Ira (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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.
Literaturempfehlungen	• Schwill, A.; Schubert, S.: Didaktik der Informatik. Berlin: Spektrum Akademischer Verlag, 2004. • Hubwieser, P.: Didaktik der Informatik. Berlin: Springer Verlag, 2000.

Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	Fachliche Grundkenntnisse der Informatik

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture period	Oral exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		2	SoSe	28
Exercises		2	SoSe	28
Präsenzzeit Modul insgesamt				56 h

mat950 - Discrete Mathematics

Module label	Discrete Mathematics
Modulkürzel	mat950
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Aufbaucurriculum - Pflichtbereich • Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule
Zuständige Personen	Heß, Florian (Module responsibility) Stein, Andreas (Module responsibility) Stein, Sandra (Module responsibility)
Prerequisites	
Skills to be acquired in this module	• 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
Literaturempfehlungen	Kreußler, Pfister: Mathematik für Informatiker, Springer 2009. Knauer, Knauer: Diskrete und algebraische Strukturen - kurz gefasst, Springer 2015. Aigner: Diskrete Mathematik, Vieweg 2006. Beutelspacher, Zschiegner: Diskrete Mathematik für Einsteiger, Vieweg 2014. Epp: Discrete Mathematics with Applications, Brooks Cole 2011. Graham, Knuth, Patashnik: Concrete Mathematics, Addison-Wesley 1994. Hartmann: Mathematik für Informatiker, Vieweg 2014. Rosen: Discrete Mathematics and its applications, McGraw-Hill 2018. Steger: Diskrete Strukturen, Band 1, Springer 2007. Teschl, Teschl: Mathematik für Informatiker, Band 1, Springer 2013. Further reading will be announced in the lecture.
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	annual
Module capacity	unlimited
Reference text	Im Zwei-Fächer Bachelor Informatik ist dieses Modul im Basiscurriculum zu studieren.
Modullevel / module level	AC (Aufbaucurriculum / Composition)
Modulart / typ of module	Pflicht / Mandatory
Lehr-/Lernform / Teaching/Learning method	
Vorkenntnisse / Previous knowledge	

Examination	Prüfungszeiten	Type of examination
Final exam of module	after the end of the lecture period	Written exam or oral exam. Bonus points can be earned.

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	WiSe	42

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Exercises		1	WiSe	14
Präsenzzeit Modul insgesamt				56 h

Aufbaumodule (60 KP)

inf005 - Software Engineering I

Module label	Software Engineering I
Modulkürzel	inf005
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Aufbaucurriculum - Pflichtbereich • Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule • Bachelor's Programme Mathematics (Bachelor) > Nebenfachmodule • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule (60 KP) • Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Pflichtbereich • Master's Programme Environmental Modelling (Master) > Mastermodule
Zuständige Personen	Winter, Andreas (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
Skills to be acquired in this module	The objective of the module is to convey the development and maintainance of large scale software systems. The complete software developing process including requirements collection, software architecture and quality control is observed. The basics of object oriented modelling and software development are enhanced. **Professional competence** The students: • comprehend the different developmental phases of software (especially requirements engineering, software design, software implementation and quality control) • name the tasks of each phase • select appropriate methodical utilities • select suitable methods and utilities for each project phase • understand the advantages of the modelling process with UML • model moderate tasks in UML • understand and develop solutions for given problems by means of development environments **Methodological competence** The students: • structure, document and evaluate problems and solutions with the tools of object oriented modelling • apply methods and techniques of object oriented modelling purposefully **Social competence** The students: • create, present and discuss solutions with modelling techniques - • present and solve modelling problems in teams **Self-competence** The students: reflect their problem-solving behaviour with regard to the capabilities of software technology
Module contents	The module introduces fundamental terms and concepts in software engineering. This includes: - need for software engineering - activities and process-models in software development - object-oriented modelling, meta modelling - Interdependencies between code and models - requirements elicitation - definition of software architectures - application of software patterns - software quality management - software maintenance, evolution and operation Software engineering tools are presented and applied in practical exercises.
Literaturempfehlungen	Ian Sommerville: Software Engineering, Addison-Wesley Longman, Amsterdam, 10. Ed. 2012 Jochen Ludewig, Horst Lichter: Software Engineering, dpunkt.verlag, 3. Auflage 2013 Helmut Balzert: Lehrbuch der Software-Technik, Spektrum Akademischer Verlag, 3. Auflage 2009 Chris Rupp, Stefan Queins: UML 2 glasklar. Praxiswissen für die UML-Modellierung, Carl Hanser Verlag, 4. Auflage 2012
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	AC (Aufbaucurriculum / Composition)
Modulart / typ of module	Pflicht / Mandatory

Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	- inf030 - inf031

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture period or during the lecture period (portfolio)	Written exam or oral exam or portfolio (? 3 services)

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	WiSe	42
Exercises		2	WiSe	28
Präsenzzeit Modul insgesamt				70 h

inf007 - Information Systems I

Module label	Information Systems I
Modulkürzel	inf007
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Aufbaucurriculum - Pflichtbereich • Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule • Bachelor's Programme Economics and Business Administration (Bachelor) > Studienrichtung Wirtschaftsinformatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule (60 KP) • Master Applied Economics and Data Science (Master) > Specialization • Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Pflichtbereich
Zuständige Personen	Lehrenden, Die im Modul (Prüfungsberechtigt) Wingerath, Wolfram (Module responsibility)
Prerequisites	
Skills to be acquired in this module	This module introduces the core concepts, languages and architectures of databases. In software systems these concepts are important. Professional competence The students: • name the core concepts of the languages and architectures of databases (especially) • select data models • integrate structuring concepts of information systems in their designs Methodological competence The students: • design database systems appropriatelyanalyse problems from the field of database-supported • information systems and solve them appropriately Social competence The students: • enhance their ability to work in a team Self-competence The students: • reflect their problem-solving behaviour with regard to the information processing concepts
Module contents	• Relational data models • Relational algebra and its implementation in SQL (the standard of databases) • Database design on different abstractions (conceptual and logical design) • Normalisation - Data base architectures • Distributed and active databases • Object-oriented, object-related and XML-based database systems
Literaturempfehlungen	• Ramez Elmasri und Shamkant B. Navathe (2016), Fundamentals of Databases Systems, 7th Revised edition, Pearson/Addison Wesley.
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	AC (Aufbaucurriculum / Composition)

Modulart / typ of module		Wahlmodul / Opportunity			
Lehr-/Lernform / Teaching/Learning method		V+Ü			
Vorkenntnisse / Previous knowledge					
Examination		Prüfungszeiten		Type of examination	
Final exam of module		At the end of the lecture period		Written or oral exam	
Form of instruction	Comment	SWS		Frequency	Workload of compulsory attendance
Lecture		3		WiSe	42
Exercises		1		WiSe	14
Präsenzzeit Modul insgesamt					56 h

inf700 - Computer Science Education I

Module label	Computer Science Education I
Modulkürzel	inf700
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Wahlbereich Informatik, Kultur und Gesellschaft • Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Bachelor's Programme Computing Science (Bachelor) > Wahlbereich Informatik, Kultur und Gesellschaft • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule (60 KP) • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule
Zuständige Personen	Diethelm, Ira (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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.
Literaturempfehlungen	• Schwill, A.; Schubert, S.: Didaktik der Informatik. Berlin: Spektrum Akademischer Verlag, 2004. • Hubwieser, P.: Didaktik der Informatik. Berlin: Springer Verlag, 2000.

Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	Fachliche Grundkenntnisse der Informatik

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture period	Oral exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		2	SoSe	28
Exercises		2	SoSe	28
Präsenzzeit Modul insgesamt				56 h

Praktische Vertiefung (60 KP)

inf004 - Software Project

Module label	Software Project
Modulkürzel	inf004
Credit points	9.0 KP
Workload	270 h
Verwendbarkeit des Moduls	

- Bachelor's Programme Biology (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Business Administration and Law (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Business Informatics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Chemistry (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Comparative and European Law (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel more...
- Bachelor's Programme Computing Science (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Economics and Business Administration (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Education (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Engineering Physics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Environmental Science (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Intercultural Education and Counselling (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Mathematics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Physics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Physics, Engineering and Medicine (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Social Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Bachelor's Programme Sustainability Economics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Art and Media (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Biology (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Chemistry (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP)
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Dutch Linguistics and Literary Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Economic Education (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Economics and Business Administration (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Education (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Elementary Mathematics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme English Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Gender Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme General Education (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme German Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme History (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Dual-Subject Bachelor's Programme Material Culture: Textiles (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel

	• Dual-Subject Bachelor's Programme Mathematics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel • Dual-Subject Bachelor's Programme Music (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel • Dual-Subject Bachelor's Programme Philosophy / Values and Norms (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel • Dual-Subject Bachelor's Programme Physics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel • Dual-Subject Bachelor's Programme Politics-Economics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel • Dual-Subject Bachelor's Programme Protestant Theology and Religious Education (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel • Dual-Subject Bachelor's Programme Slavic Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel • Dual-Subject Bachelor's Programme Social Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel • Dual-Subject Bachelor's Programme Special Needs Education (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel • Dual-Subject Bachelor's Programme Sport Science (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel • Dual-Subject Bachelor's Programme Technology (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel • Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
Zuständige Personen	Grawunder, Marco (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	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.
Literaturempfehlungen	
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 / module level	BC (Basiscurriculum / Base curriculum)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V + Ü + PR
Vorkenntnisse / Previous knowledge	

Examination	Prüfungszeiten	Type of examination
Final exam of module	by appointment	Portfolio

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		2	WiSe	28
Exercises		2	SoSe und WiSe	28
Project		4	SoSe und WiSe	56
Präsenzzeit Modul insgesamt				112 h

inf009 - Database Practical

Module label	Database Practical
Modulkürzel	inf009
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik • Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Praktische Informatik) • Master of Education Programme (Hauptschule and Realschule) Computing Science (Master of Education) > Mastermodule • Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik
Zuständige Personen	Grawunder, Marco (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
Skills to be acquired in this module	The objective of this module is to gather practical experience on databases and information systems. The students get an overview of the technical realisation, implementation and optimisation of a professional database management system. **Professional competence** The students: • Realise, implement and program data base systems • Program and implement database-oriented system routines • Implement optimisation goals in the modelling phase • Administer professional database systems (installation, maintenance and adjustment) • Recognise database systems' performance problems and are able to fix them with according methods • Organise and control processes of database systems **Social competence** The students: • Solve database system problems in a team **Self-competence** The students: • Acknowledge the limits of their ability to cope with pressure during the implementation and are aware of failures • Reflect their self-perception
Module contents	The module “Practical Course Databases” is a related practical course of the module “Information Systems I”. The objectives of this module are special technical concepts of a database system and practical solutions in database programming and optimisation. Contents of this module are: • System-oriented database management programming, • Implementation of catalogue systems, • Optimisation strategies based on parallelisation and partitioning requirements
Literaturempfehlungen	Ramez Elmasri und Shamkant B. Navathe (2007). Fundamentals of Databases Systems. Fifth Edition, Pearson/Addison Wesley. Held Andrea (2005), Oracle 10g Hochverfügbarkeit Addison-Wesley. Held Andrea (2015), Oracle 12c New Features Addison Wesley. Feuerstein Steven, Pribyl Bill, Dawes Chip (2007). Oracle PL/SQL. 4. Auflage, O'Reillys Taschenbibliothek
Links	http://www-is.informatik.uni-oldenburg.de/227/

Language of instruction	German	
Duration (semesters)	1 Semester	
Module frequency	jährlich	
Module capacity	unlimited	
Modullevel / module level	AS (Akzentsetzung / Accentuation)	
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht	
Lehr-/Lernform / Teaching/Learning method	P	
Vorkenntnisse / Previous knowledge	Informationssysteme I Betriebssystemkenntnisse	
Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture period	Oral exam
Form of instruction	Practical training	
SWS	4	
Frequency	WiSe	
Workload Präsenzzeit	56 h	

inf014 - Operating Systems Practical

Module label	Operating Systems Practical
Modulkürzel	inf014
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik • Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP) • Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik
Zuständige Personen	Theel, Oliver (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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
Literaturempfehlungen	Patterson and Hennessy, Computer Organization and Design, 3rd edition, Morgan Kaufmann, 2007

Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Reference text	**Associated with the modules:** • Betriebssysteme I • Betriebssysteme II • Verteilte Systeme
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	P
Vorkenntnisse / Previous knowledge	- Betriebssysteme I - Betriebssysteme II - Programmiersprachen: C, Assembler

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the semester	Active participation / work report and oral exam

Form of instruction	Practical training
SWS	4
Frequency	WiSe
Workload Präsenzzeit	56 h

inf018 - Media Processing

Module label		Media Processing
Modulkürzel		inf018
Credit points		6.0 KP
Workload	180 h	
Verwendbarkeit des Moduls		

- Bachelor's Programme Biology (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Business Administration and Law (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Business Informatics (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik
- Bachelor's Programme Business Informatics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Chemistry (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer" more...
- Bachelor's Programme Comparative and European Law (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Bachelor's Programme Computing Science (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Economics and Business Administration (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Education (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Engineering Physics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Environmental Science (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Intercultural Education and Counselling (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Mathematics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Physics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Physics, Engineering and Medicine (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Social Studies (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Bachelor's Programme Sustainability Economics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Art and Media (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Biology (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Chemistry (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP)
- Dual-Subject Bachelor's Programme Dutch Linguistics and Literary Studies (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Economic Education (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Economics and Business Administration (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Education (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Elementary Mathematics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme English Studies (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Gender Studies (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme General Education (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme German Studies (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme History (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Material Culture: Textiles (Bachelor) > PP "Medieninformatik für

Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Mathematics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Music (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Philosophy / Values and Norms (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Physics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Politics-Economics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Protestant Theology and Religious Education (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Slavic Studies (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Social Studies (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Special Needs Education (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Sport Science (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Dual-Subject Bachelor's Programme Technology (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Praktische Informatik)
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik
- Master's Programme Business Informatics (Master) > Akzentsetzungsmodule der Informatik

Zuständige Personen

Boll-Westermann, Susanne (Module responsibility)

Lehrenden, Die im Modul (Prüfungsberechtigt)

Prerequisites

Skills to be acquired in this module

The students can explain the basics of image processing and know which algorithms exist for the basic tasks in image processing and how these are applied.

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

Professional compentence:
The students

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

Methodological competence:
The students

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

Social competence
The students:

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

Self competence
The students:

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

Module contents	The lecture covers the technologies of media processing. In particular, the lecture focuses on image processing chain from digital imaging, through image pre-/and postprocessing, and image storage to image analysis. In addition to compression techniques and color space theory (RGB, HSV, YUV, CIEXYZ, ...), the topics of the lecture include image enhancement, feature extraction, feature description, image analysis and image comprehension. The lecture furthermore discusses the encoding and analysis of video and audio.
Literaturempfehlungen	Wilhelm Burger und Mark James Burge. Digitale Bildverarbeitung: Eine Einführung mit Java und Image, J. Springer, 2006. Literatur im Handapparat der Abteilung in der Bibliothek. Linkliste im Lernmanagementsystem zu den einzelnen Themen der Vorlesung.
Links	http://medien.informatik.uni-oldenburg.de/lehre
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	Annually
Module capacity	unlimited
Reference text	Useful previus knowlodge: Solid programming skills in Java and/or C++, practical informatics. Interest in media processing
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	1V + 1Ü
Vorkenntnisse / Previous knowledge	Gute Programmierkenntnisse in PythonJava und/oder JavaC++, Interesse an Medienverarbeitung.

Examination	Prüfungszeiten	Type of examination
Final exam of module	Die Vorstellung des praktischen Projektes an einem Projekttag aller Kleingruppen findet direkt im Anschluss an die Vorlesungszeit statt. Die mündliche Prüfung findet in den ersten beiden Wochen nach Ende der Vorlesungszeit statt. Etwaige Nachprüfungen finden am Ende der vorlesungsfreien Zeit statt. Der genaue Zeitplan kann den Webseiten der Abteilung sowie den Angaben im Lernmanagementsystem Stud.IP entnommen werden.	Project and oral exam The portfolio comprises two graded submodules: • Practical group project which progress has to be presented regularly during the tutorials. • Oral exam on the topics of the lecture. Practical project and oral exam count 50% each to the final grade. Both practical project and oral exam must be passed individually.

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		2	WiSe	28
Project		2	WiSe	28
Präsenzzeit Modul insgesamt				56 h

inf021 - Advanced Java Technologies

Module label	Advanced Java Technologies
Modulkürzel	inf021
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik • Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP) • Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik
Zuständige Personen	Boles, Dietrich (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
Skills to be acquired in this module	The objective of this module is to introduce advanced concepts and technologies of the Java Standard Edition. The students will be able to use the technologies to implement large applications. **Professional competence:** The students: - name the essential packages of the JDK class library - structure large programs properly and implement them extensively - look up required classes in the JDK-Library and solve problems with these classes - structure their programs properly - understand and interpret large programs of other students - evaluate the quality of large programs related to their maintainability, reuseability and expandability **Methodological competence:** The students: - search for solutions to specific problems in the internet independently **Social competence:** The students: - discuss own and solutions of other students **Self-competence:** The students: - reflect their problem-solving behaviour and take up new solutions, e.g. from the internet
Module contents	A selection of the following subjects is presented during the lectures: - GUI (AWT, Swing, JavaFX) - Java-Basics and Collection-API - Graphics and multimedia - Events - Model-View-Control (MVC) - Threads - Internationalization, localization - Reflection - IO, Files - Tools (compiler, classloader, printer, ...) - Storage technologies (XML and serialization) - Distributed programming (sockets and RMI) - Databases (JDBC) - Compression - Security concepts Alternatively, a single topic is explored in depth. As part of the exercises, individual programming tasks or a larger programming task will be worked on. The tasks are related to the topic of the individual lecture contents.

Literaturempfehlungen	list of links in the learning management system
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	every Semester
Module capacity	unlimited
Reference text	
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	Wahlpflicht / Elective
Lehr-/Lernform / Teaching/Learning method	VL + Ü
Vorkenntnisse / Previous knowledge	Objektorientierte Programmierung

Examination	Prüfungszeiten	Type of examination
Final exam of module	throughout the semester	practical exercises As part of the exercises, the students work on practical programming tasks. For this purpose, new subtasks with reference to the respective lecture content have to be worked on weekly.

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		2	SoSe oder WiSe	56
Exercises		2	SoSe oder WiSe	28
Präsenzzeit Modul insgesamt				84 h

inf202 - Computer Engineering Practical

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

Verwendbarkeit des Moduls

- Bachelor's Programme Biology (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Business Administration and Law (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Business Informatics (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Chemistry (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Comparative and European Law (Bachelor) > Fachnahe Angebote Informatik

more...

- Bachelor's Programme Computing Science (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Economics and Business Administration (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Education (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Engineering Physics (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Environmental Science (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Intercultural Education and Counselling (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Mathematics (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Physics (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Physics, Engineering and Medicine (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Social Studies (Bachelor) > Fachnahe Angebote Informatik
- Bachelor's Programme Sustainability Economics (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Art and Media (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Biology (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Chemistry (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP)
- Dual-Subject Bachelor's Programme Dutch Linguistics and Literary Studies (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Economic Education (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Economics and Business Administration (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Education (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Elementary Mathematics (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme English Studies (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Gender Studies (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme General Education (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme German Studies (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme History (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Material Culture: Textiles (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Mathematics (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Music (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Philosophy / Values and Norms (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Physics (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Politics-Economics (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Protestant Theology and Religious Education (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Slavic Studies (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Social Studies (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Special Needs Education (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Sport Science (Bachelor) > Fachnahe Angebote Informatik
- Dual-Subject Bachelor's Programme Technology (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft (Bachelor) > Fachnahe Angebote Informatik
- Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik)
- Master of Education Programme (Hauptschule and Realschule) Computing Science (Master of Education) > Mastermodule
- Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik

Zuständige Personen

Fränzle, Martin Georg (Module responsibility)

Janßen, Detlef (Module responsibility)

	Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	Empfehlung: inf200 „Grundlagen der Technischen Informatik
Skills to be acquired in this module	Diese Veranstaltung versetzt die Studierenden in die Lage, informationstechnische Systeme zu analysieren, einzelne Komponenten von Rechnern zu verstehen, sie zu entwerfen und zu optimieren sowie qualifiziert über domänspezifischen Hardwareentwurf zu diskutieren. **Fachkompetenz:** Die Studierenden • beschreiben einzelne Komponenten von Rechnern • entwerfen und optimieren einzelne Komponenten von Rechnern • entwerfen und optimieren Automaten • spezifizieren und implentieren autonome Systeme **Methodenkompetenz** Die Studierenden • synthetisieren Rechnerarchitekturen • können Methoden des Hardwareentwurfs auf verschiedene Systeme transferieren **Sozialkompetenz** Die Studierenden • diskutieren qualifiziert über Hardware **Selbstkompetenz** Die Studierenden • sind dazu in der Lage, ihren Kenntnisstand klar gegen Fachkräfte verwandter Disziplinien abzugrenzen
Module contents	Dieses Modul ist der praktische Teil der Veranstaltung Einführung in die Technische Informatik
Literaturempfehlungen	Skript zur Veranstaltung, Patterson, D.A., Hennesy, J.L.:Computer Organisation and Design: The Hardware/Software Interface
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	Jedes Sommersemester
Module capacity	unlimited
Modullevel / module level	AC (Aufbaucurriculum / Composition)
Modulart / typ of module	Wahlpflicht / Elective
Lehr-/Lernform / Teaching/Learning method	P
Vorkenntnisse / Previous knowledge	

Examination	Prüfungszeiten	Type of examination
Final exam of module	Am Ende der Vorlesungszeit	PK

Form of instruction	Practical training
SWS	4
Frequency	SoSe
Workload Präsenzzeit	56 h

inf406 - Laboratory Real-Time Systems

Module label	Laboratory Real-Time Systems
Modulkürzel	inf406
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP) • Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik
Zuständige Personen	Olderog, Ernst-Rüdiger (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	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.
Literaturempfehlungen	E.-R. Olderog, H. Dierks: Real-Time Systems: Formal Specification and Automatic Verification, Cambridge University Press, 2008
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	unregelmäßig
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht

Lehr-/Lernform / Teaching/Learning method	P	
Vorkenntnisse / Previous knowledge		
Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture period	exercises
Form of instruction	Practical training	
SWS	4	
Frequency	WiSe	
Workload Präsenzzeit	56 h	

inf803 - Special Topics in Computer Science I

Module label	Special Topics in Computer Science I
Modulkürzel	inf803
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik • Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP) • Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik
Zuständige Personen	Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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
Literaturempfehlungen	According to the assigned task
Links	
Languages of instruction	German, English
Duration (semesters)	1 Semester
Module frequency	halbjährlich
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	2 Veranstaltungen aus V, Ü, S, P, PR
Vorkenntnisse / Previous knowledge	

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

Form of instruction	VA-Auswahl

SWS	4
Frequency	SoSe oder WiSe
Workload Präsenzzeit	56 h

inf804 - Special Topics in Computer Science II

Module label	Special Topics in Computer Science II
Modulkürzel	inf804
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik • Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP) • Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik
Zuständige Personen	Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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
Literaturempfehlungen	According to the assigned task
Links	
Languages of instruction	German, English
Duration (semesters)	1 Semester
Module frequency	halbjährlich
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	2 Veranstaltungen aus V, Ü, S, P, PR
Vorkenntnisse / Previous knowledge	

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

Form of instruction	VA-Auswahl

SWS	4
Frequency	SoSe oder WiSe
Workload Präsenzzeit	56 h

inf808 - Current Topics in Computer Science

Module label	Current Topics in Computer Science
Modulkürzel	inf808
Credit points	3.0 KP
Workload	90 h
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik • Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP) • Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik
Zuständige Personen	Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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
Literaturempfehlungen	According to the assigned task
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	unregelmäßig
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	1 Veranstaltung aus V, Ü, S, P, PR
Vorkenntnisse / Previous knowledge	

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

Examination	Prüfungszeiten	Type of examination
		exam
Form of instruction	VA-Auswahl	
SWS	2	
Frequency	SoSe oder WiSe	
Workload Präsenzzeit	28 h	

inf800 - Proseminar in Computer Science

Module label	Proseminar in Computer Science		
Modulkürzel	inf800		
Credit points	3.0 KP		
Workload	90 h		
Verwendbarkeit des Moduls	• Bachelor's Programme Business Informatics (Bachelor) > Aufbaucurriculum - Pflichtbereich • Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Praktische Vertiefung (60 KP) • Master of Education Programme (Vocational and Business Education) Computing Science (Master of Education) > Praktische Vertiefung der Informatik		
Zuständige Personen	Diethelm, Ira (Module responsibility) Nieße, Astrid (Module responsibility) Sauer, Jürgen (Module responsibility) Lehrenden, Die im Modul (Module counselling)		
Prerequisites	Studierende im den Bachelor-Studiengängen der Informatik sowie Master of Education Informatik		
Skills to be acquired in this module	Supported by a lecturer the students familiarise with a given topic by literature research. They understand and evaluate the relevance of the literature. After this evaluation the students present and discuss their solutions academically. **Professional competence**: The students: - Characterise and apply computer science basics (algorithms, data structures, programming, basics of practical, technical and theoretical computer science) - Define und describe essential mathematical, logical and physical basics of computer science - Define and illustrate the core disciplines of computer science (theoratical, practical and technical computer science) **Methodological competence**: The students: - Examine problems, use formal methods to phrase them and analyze them appropriately - Evaluate problems by the use of technical and scientific literature - Reflect on a scientific topic and write a scientific seminar paper under guidance and present their findings **Social competence** The students: - Communicate considerately and appropriately with users and experts - Use presentation methods **Self-competence** The students: - Plan their informatical actions independently - Reflect their contributions critically and discuss them with users and experts - Collect and update their knowledge independently		
Module contents	according to the assigned task		
Literaturempfehlungen	according to the assigned task		
Links			
Language of instruction	German		
Duration (semesters)	1 Semester		
Module frequency	halbjährlich		
Module capacity	unlimited		
Reference text	Choose one of the seminaire courses of the module.		
Modullevel / module level			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht		
Lehr-/Lernform / Teaching/Learning method	S		
Vorkenntnisse / Previous knowledge			
Examination	Prüfungszeiten	Type of examination	
Final exam of module	Am Ende des Semesters und nach Absprache	Presentation	
Form of instruction	Seminar		
SWS	2		

Frequency	SoSe oder WiSe
Workload Präsenzzeit	28 h

Wahlpflicht Technische Informatik (30 KP)

inf200 - Computer Engineering I

Module label	Computer Engineering I
Modulkürzel	inf200
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Basismodule • Bachelor's Programme Mathematics (Bachelor) > Nebenfachmodule • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Basismodule • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
Zuständige Personen	Nebel, Wolfgang (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
Skills to be acquired in this module	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 analyse 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 t 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.
Literaturempfehlungen	- handout manuscript of the course - Schiffmann, W.; Schmitz, R. (2001): Technische Informatik I, II, Übungsbuch; Springer Verlag, Berlin. - Dal Cin, M. (1996): Rechnerarchitektur; B.G. Teubner. - Lagemann, K. (1987): Rechnerstrukturen; Springer-Verlag, Berlin. - Oberschelp, W.; Vossen, G. (1989): Rechneraufbau und Rechnerstrukturen; Oldenbourg-Verlag. - Mano, Morris M.(1993): Computer System Architecture 3; Prentice Hall. - Gajski, D.(1997): Principles of Digital Design; Prentice Hall. - Patterson, D.A.; Hennessy, J.L. (1997): Computer Organization and Design: - The Hardware/Software Interface; 2. Edition; Morgan Kaufmann Publishers. - Wilkinson, B. (1996): Computer Architecture Design and Performance; 2. Edition; Prentice Hall. - Tannenbaum, A.S.(1999): Structured Computer Organization; 4. Edition; Prentice Hall.
Links	
Language of instruction	German
Duration (semesters)	1 Semester

Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	BC (Basiscurriculum / Base curriculum)
Modulart / typ of module	
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the semester	Written or oral exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	WiSe	42
Exercises		1	WiSe	14
Präsenzzeit Modul insgesamt				56 h

inf201 - Computer Engineering II

Module label	Computer Engineering II
Modulkürzel	inf201
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik) • Master of Education Programme (Hauptschule and Realschule) Computing Science (Master of Education) > Mastermodule
Zuständige Personen	Nebel, Wolfgang (Module responsibility) Wüstefeld, Manuela (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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 iIn 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.
Literaturempfehlungen	- Lecture notes - Oberschelp, W., Vossen, G.: Rechneraufbau und Rechnerstrukturen; Oldenbourg Verlag - Gajski, D.: Principles of Digital Design; Prentice Hall 1997 - Patterson, D.A., Hennesy, J.L.: Computer Organisation and Design: The Hardware/Software Interface; 2. Edition; Morgan Kaufman Publishers, 1997 - Tannenbaum, A.S.: Structured Computer Organization ; 4. Edition; Prentice Hall, 1999 Additional literature will be mentioned in the lectures
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	AC (Aufbaucurriculum / Composition)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture period	Written or oral Exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	SoSe	42
Exercises		1	SoSe	14

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Präsenzzeit Modul insgesamt				56 h

inf203 - Embedded Systems I

Module label	Embedded Systems I
Modulkürzel	inf203
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik)
Zuständige Personen	Nebel, Wolfgang (Module responsibility) Fränzle, Martin Georg (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
Skills to be acquired in this module	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 excercises self-responsibly
Module contents	Embedded systems support complex feedback problems, control problems and data processing tasks. They have an important value creation potential for telecommunications, production management, transport and electronics. The functionality of embedded systems is realised by the integration of processors, special hardware and software. The embedded systems design is influenced by the heterogeneity of system architectures, the complexity of systems and technical and economic requirements. This module gives an overview of embedded systems and their design. The process of digital signals is especially important for telecommunications and multimedia. For this purpose, the module introduces digital signal processing algorithms. The principles of feedback control are introduced by exemplary transport applications. Subsequently, the module provides the specifications and language characteristics of the embedded system design. For this purpose, graphical data-flow modelling languages (for instance Simulink) and control-flow specifications (for instance State Charts) are presented. The module closes with the concepts of possible architectures and communication models. Hands-on exercises with the tools Matlab/Simulink/StateFlow support the module contents.
Literaturempfehlungen	**Slides and:**

- Harel, D.: STATECHARTS: A Visual Formalism for Complex Systems. Science of Computer Programming, 8, North-Holland, 1987, page(231-274)
- Harel D.: Naamad, A. The STATEMATE Semantics of Statecharts. ACM Trans. Software Engineering Methods, Oct 1996
- Harel, D.; Politi, M.: Modeling Reactive Systems with Statecharts: The Statemate Approach
- Josef Hoffmann: Matlab und Simulink: Beispielorientierte Einführung in die Simulation dynamischer Systeme, Addison-Wesley, 1998, ISBN 3-8273-1077-6
- Staunstrup, J., Wolf, W. (eds.): Hardware/Software Co-Design: Principles and Practice. Kluwer Academic Publishers, 1997, ISBN 0-7923-8013-4, chapters 1, 2, (3), 4, 6, (7), (8-10)
- U. Reimers. Digitale Fernsehtechnik. 2. Aufl., Springer, 1997, ISBN 3-540-60945-8

Secundary literature:

- Debardelaben, J.A.; Gadient, A.J.: Incorporating Cost Modeling in Embedded-System Design. IEEE Design & Test, vol 13, no. 3, 1997
- De Micheli, G.; Sami, M.: Hardware-Software Co-Design. Kluwer, 1996, ISBN 0-7923-3883-9
- Gajski, D.; Vahid, F.; Narayan, S.; Gong, J.: Specification and Design of Embedded Systems. Prentice Hall, 1994, ISBN 0-13-150731-1
- T. Painter, A. Spanias. Perceptual Coding of Digital Audio. Proceedings of the IEEE, vol 88, no 4, April 2000.
- U. Freyer. DVB - Digitales Fernsehen. Verlag Technik, 1997, ISBN 3-341-01192-7
- B. Friedrichs. Kanalcodierung: Grundlagen und Anwendungen in modernen Kommunikationssystemen. Springer, 1995, ISBN 3-540-58232-0
- G.C. Clark. Error-correction coding for digital communications. 3rd printing, Plenum Press, 1988, ISBN 0-306-40615-2
- Artikelserie zum MPEG-2-Standard 3/94 - 10/94 und das Tutorial "Digitale Bildcodierung" 1/92 - 1/93, beides in "Fernseh- und Kinotechnik" (BIS: Z elt ZA 1536)

Links	
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". **Associaltes with the modules:** In the module "Eingebettete Systeme II" additonal 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 / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	- Grundlagen der technischen Informatik - Technische Informatik

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the semester	Written or oral exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	WiSe	42
Exercises		1	WiSe	14
Präsenzzeit Modul insgesamt				56 h

inf204 - Embedded Systems II

Module label	Embedded Systems II
Modulkürzel	inf204
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik)
Zuständige Personen	Nebel, Wolfgang (Module responsibility) Fränzle, Martin Georg (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
Skills to be acquired in this module	The module provides an introduction to digital embedded systems design. **Professional competence:** The students: • name embedded systems architectures • name specific hardware components and -architecture designs, particularly processor designs • characterise the design spaces and associated embedded systems design techniques • decompose subcomponents of feedback control systems and implement their tasks in different design spaces • develop software-/hardware components • describe fault-tolerance architecture principles • describe real-time and safety requirements analysing techniques • characterise hardware synthesis **Methodological competence:** The students: • estimate the consequences of design decisions in terms of energy usage, performance and reliability component allocations, and designs • implement an embedded hardware-/software system according to a given specification • model hardware with a hardware description languages • analyze Hardware-/Software systems using event-bases simulation **Social competence:** The students: • implement solutions to given problems in teams • present results of computer science problems to groups • organize themselves as a team to solve a larger problem using project management methods **Self-competence:** The students: • acknowledge the limits of their ability to cope with pressure during the implementation process of systems • deal self responsibly with exercises
Module contents	Embedded systems support complex feedback problems, control problems and data processing tasks. They have an important value creation potential for telecommunications, production management, transport and electronics. The functionality of embedded systems is realised by the integration of processors, special hardware and software. The embedded systems design is influenced by the heterogeneity of system architectures, the complexity of systems and technical and economic requirements. This module is the continuation of the module “Eingebettete Systeme I” and deals with different architectures of embedded systems and processors. The module provides system partitioning methods and the synthesis of hardware components.

	Hands-on exercises with development tools, hardware description languages and simulation support the module contents.
Literaturempfehlungen	**Slides and:** • Staunstrup, J.: Wolf, W. (eds.): Hardware/Software Co-Design: Principles and Practice. Kluwer Academic Publishers, 1997, ISBN 0-7923-8013-4, chapters 1, 2, (3), 4, 6, (7), (8-10) • Yen, Ti-Yen; Wolf, W.: Hardware-Software Co-Synthesis of Distributed Embedded Systems. Kluwer, 1996, ISBN 0-7923-9797-5 **Secondary literature:** • Peter J. Ashenden: The Designer's Guide to VHDL. Morgan Kaufmann Publishers, 2002, ISBN 1-55860-674-2 • Lehmann, G.; Wunder, B.; Selz, M.: Schaltungsdesign mit VHDL. Franzis Verlag, 1994, ISBN 3-7723-6163-3 • J. Reichardt, B. Schwarz: VHDL-Synthese, Entwurf digitaler Schaltungen und Systeme. Oldenbourg Wissenschaftsverlag, 2000, ISBN 3-486-25128-7 • Mermet, J. (ed.): Fundamentals and Standards in Hardware Description Languages. Kluwer, 1993, ISBN 0-7923-2513-3 • De Micheli, G.; Sami, M.: Hardware-Software Co-Design. Kluwer, 1996, ISBN 0-7923-3883-9 • Gajski, D.; Vahid, F.; Narayan, S.; Gong, J.: Specification and Design of Embedded Systems. Prentice Hall, 1994, ISBN 0-13-150731-1
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 / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture times	Written or oral Exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	SoSe	42
Exercises		1	SoSe	14
Präsenzzeit Modul insgesamt				56 h

inf205 - Formal Methods in Embedded System Design

Module label	Formal Methods in Embedded System Design
Modulkürzel	inf205
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik)
Zuständige Personen	Fränzle, Martin Georg (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
Skills to be acquired in this module	The module provides an overview over semantic models for reactive systems, real-time systems ,and hybrid systems, as well as examples of corresponding specification logics. It explains state-exploratory verification procedures in both explicit and symbolic variants. The knowledge acquired can be employed in all domains requiring the development of reliable software and hardware systems is concerned Professional competences The students: • make a sound judgement on the scope of the certificates that can be obtained with formal methods • assess the suitability of available verification tools for a particular problem and system class • use automatic analysis tools on real systems, interpret the results obtained and subsequently improve the system under investigation in an informed and targeted manner • prepare system models for automatic analysis procedures and encode them symbolically (or otherwise) • design and implement their own verification algorithms Methological competences The students: • master the mathematical modelling of complex and heterogeneous dynamical systems • know relevant mathematical models of dynamic systems and can instantiate them to new problem classes Social competences The students: • together in a team develop and implement basic algorithms of automatic verification • discuss the advantages and disadvantages of algorithmic alternatives and different formalisations Self competences The Students: • can assess their technical and methodological understanding • reflect on their problem-solving competence with reference to the procedures and methods presented
Module contents	Embedded computer systems maintain constant interaction with their environment. This induces interaction sequences that are difficult to predict, which complicates the construction and validation of such systems. Akin to the use of structural models for rigorous validation of building layouts in the construction industry, formal models in computer science have consequently been developed for analysis of various aspects of computing systems in general and embedded systems in particular. They cover execution time, energy demand, or possible system dynamics of embedded systems. They represent the respective aspect of the system in a closed form and thus allow for the - often fully automatic - derivation of reliable certificates that apply to any interaction scenario with the environment. This is in contrast to methods of testing or profiling, which only test selected scenarios and thus can only provide limited coverage. In this module, various such models are explained and methods for their fully automatic analysis -i.e., derivation certificates- or synthesis -i.e., automatic generation of correct system designs- from such models are explained and demonstrated in their application.

	The exercises provide opportunity for deepening understanding through hands-on experience with domain-specific modelling and verification tools, as well as by creating a (small) fully automated verification tool yourself in a guided process. In the lectures, the semantic, logical, and algorithmic basics of automatic analysis of embedded software systems are taught. The primary form of instruction is the media-supported lecture as well as the didactic question-answer game,. In the exercise classes, the knowledge acquired in the lecture is deepened and put into practice. For this purpose, in the first half of the semester, exercises are set fortnightly, the completion of which in small groups encourages independent testing of the individual understanding of the topic and peer teaching. In the second half of the semester, a larger tool development task is set, also to be pursued on in small groups of 3 students each. The work on these projects spans the entire second half of the semester and offers the possibility of project-oriented learning. In this phase, the exercise classes serve as consultation time with the lecturers; in particular, solution approaches and problems can be presented and discussed.
Literaturempfehlungen	• Michael Huth, Mark Ryan: Logic in Computer Science: Modelling and Reasoning About Systems. Cambridge University Press, 2004. • Christel Baier, Joost-Pieter Katoen: Principles of Model Checking. MIT Press, 2008. • Edmund M. Clarke, Orna Grumberg, Doron A. Peled: Model Checking. MIT Press, 2000.
Links	
Languages of instruction	German, English
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	Fundierte Grundkenntnisse in mathematischer Logik, diskreter Mathematik, Automaten- und Berechenbarkeitstheorie, wie sie in den Modulen "Diskrete Strukturen" und "Theoretische Informatik I + II" vermittelt werden. Zudem Programmierkenntnisse, wie sie im "Programmierkurs" erworben werden. Begründung: Die in der Vorlesung vorgestellten Verfahren basieren auf einer Operationalisierung von Semantik durch Reduktion auf logische Kodierungen und mechanisierte Prüfung logischer Aussagen. Ein Verständnis dieser Inhalte sowie ihre werkzeugtechnische Umsetzung bedarf der Grundlagen aus den vorgenannten Veranstaltungen.

Examination	Prüfungszeiten	Type of examination
Final exam of module	• 1st deadline: Submission of the semester project incl. one week after the end of the lecture period of the lecture period; followed by a colloquium and a final discussion • 2nd deadline: Repeat of the submission of the semester project incl. written elaboration two weeks before the beginning of the following semester followed by colloquium and final discussion	Semester project

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	WiSe	42
Exercises		1	WiSe	14
Präsenzzeit Modul insgesamt				56 h

inf207 - Electrical Engineering

Module label	Electrical Engineering
Modulkürzel	inf207
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik) • Master of Education Programme (Hauptschule and Realschule) Computing Science (Master of Education) > Mastermodule • Master's Programme Computing Science (Master) > Interdisziplinäre Module
Zuständige Personen	Hein, Andreas (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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
Literaturempfehlungen	**essential:** • slides • Albach: Grundlagen der Elektrotechnik 1 und 2. Pearson Studium, 2004. **recommended:** • Hagmann, G.: Grundlagen der Elektrotechnik. AULA-Verlag, 2002.

- Hagmann, G.: Aufgabensammlung zu den Grundlagen der Elektrotechnik. AULA-Verlag, 2002.

Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	Modul Analysis II oder Numerik

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the End of the Semester	Hands-on exercises / written exam or oral exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	SoSe	42
Exercises		1	SoSe	14
Präsenzzeit Modul insgesamt				56 h

inf208 - Microrobotics and Microsystems Technology

Module label	Microrobotics and Microsystems Technology
Modulkürzel	inf208
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik) • Master's Programme Computing Science (Master) > Interdisziplinäre Module
Zuständige Personen	Fatikow, Sergej (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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, electrorheological 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
Literaturempfehlungen	**Essential:**

- Lecture notes

Recommended:

- Fatikow, S.: Mikroroboter und Mikromontage, Teubner, Stuttgart Leipzig, 2000
- Fatikow, S./Rembold, U.: Microsystem Technology and Microrobotics, Springer, Berlin Heidelberg New York, 1997

Secondary Literature (only available for some subareas!):

- Brück, A. und Schmidt, A.: Angewandte Mikrotechnik, Hanser, München Wien, 2001
- Ehrfeld, W. (Hrsg.): Handbuch Mikrotechnik, Hanser, München Wien, 2000
- Elbel, Th.: Mikrosensorik, Vieweg, Wiesbaden, 1996
- Fukuda, T. and Menz, W. (Eds.): Micro Mechanical Systems, Elsevier, Amsterdam, 1998
- Gardner, J.W.: Microsensors, Wiley, Chichester, 1994
- Gerlach, G. und Dötzel, W.: Grundlagen der Mikrosystemtechnik, Hanser, München Wien, 1997
- Krause, W.: Fertigung in der Feinwerk- und Mikrotechnik, Hanser, 1995
- Menz, W. und Mohr, J.: Mikrosystemtechnik für Ingenieure, VCH, Weinheim, 1997
- Mescheder, U.: Mikrosystemtechnik, Teubner, Stuttgart Leipzig, 2000
- Tränkler, H.-R. und Obermeier, E. (Hrsg.): Sensortechnik, Springer, Berlin Heidelberg, 1998
- Völklein, F. und Zetterer, Th.: Einführung in die Mikrosystemtechnik, Vieweg, Wiesbaden, 2000

Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Reference text	**Associated with the modules:** Embedded Systems and Microrobotics
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	Analysis II oder Numerik

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the semester	Oral exam in German

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	WiSe	42
Exercises		1	WiSe	14
Präsenzzeit Modul insgesamt				56 h

inf209 - Control Theory

Module label	Control Theory
Modulkürzel	inf209
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik) • Master's Programme Computing Science (Master) > Interdisziplinäre Module
Zuständige Personen	Fatikow, Sergej (Module responsibility) Hein, Andreas (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	• Module Differential Equations • Module Basics Electrical Engeneering
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 stabilty and determine their optimal control parameters **Methodological competence** The students: • Are aware of the technical challenges and solve them by including the implementations of other disciplines and methods **Social competence** The students: • Present solutions for specific questions **Self-competence** The students: • Get used to the specific challenges of the development of controlled systems
Module contents	Basics; analog transfer elements: linear time invariant (LTI-) systems; simulation and modeling; step response; frequency response; frequency response locus; differential equations and transfer function; control loop stability; types of controlled systems; types of linear controllers; linear control loops: reference and disturbance reaction of the controlled system; rules for control loop optimization; methods of analysis and synthesis, implementation; computerbased control MATLAB/Simulink
Literaturempfehlungen	• Unbehauen, H.:Regelungstechnik I, Klassische Verfahren zur Analyse und Synthese linearer kontinuierlicher Regelsysteme • Lutz, H. und Wendt, W.:Taschenbuch der Regelungstechnik • further reading will be announced at lecture
Links	

Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	- Differenzialgleichungen - Analysis II - Grundlagen der Elektrotechnik

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture period	Hands-on exercises and written or oral exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	WiSe	42
Exercises		1	WiSe	14
Präsenzzeit Modul insgesamt				56 h

inf210 - Signal and Image Processing

Module label	Signal and Image Processing
Modulkürzel	inf210
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Technische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Technische Informatik) • Master's Programme Computing Science (Master) > Interdisziplinäre Module
Zuständige Personen	Hein, Andreas (Module responsibility) Fränzle, Martin Georg (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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
Literaturempfehlungen	**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.; Pepges, R.; Bildverarbeitung in der Medizin; Springer Verlag 1997 -Handels. H.; Medizinische Bildverarbeitung; Teubner Verlag, Stuttgart - Leipzig 2000
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	Modul math040 Analysis II b: Differentialrechnung mehrerer Variablen

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the semester	Hands-on exercises and written or oral exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		2	WiSe	28
Exercises		2	WiSe	28
Präsenzzeit Modul insgesamt				56 h

Wahlpflicht Theoretische Informatik (30 KP)

inf400 - Theoretical Computer Science: Logic

Module label	Theoretical Computer Science: Logic
Modulkürzel	inf400
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Basismodule • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
Zuständige Personen	Olderog, Ernst-Rüdiger (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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

Literaturempfehlungen	Essential: • Script "Logik" (in German), in its current edition Recommended: • D. van Dalen: "Logic and Structure", Fourth Edition. Springer-Verlag, 2004. Good secondary reading: • U. Schöning: "Logic for Computer Scientists", Birkhäuser, 1994.

Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	BC (Basiscurriculum / Base curriculum)
Modulart / typ of module	Pflicht / Mandatory
Lehr-/Lernform / Teaching/Learning method	V & Ü
Vorkenntnisse / Previous knowledge	

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture period	written exam or oral exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	SoSe	42
Exercises		1	SoSe	14
Präsenzzeit Modul insgesamt				56 h

inf401 - Foundations of Theoretical Computer Science

Module label	Foundations of Theoretical Computer Science
Modulkürzel	inf401
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Aufbaumodule • Bachelor's Programme Mathematics (Bachelor) > Nebenfachmodule • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Pflichtmodule • Master of Education Programme (Hauptschule and Realschule) Computing Science (Master of Education) > Mastermodule
Zuständige Personen	Olderog, Ernst-Rüdiger (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
Skills to be acquired in this module	Introduction to the theory of automata, formal languages, computability, and complexity **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.
Literaturempfehlungen	- essentiell: Skript "Grundbegriffe der Theoretischen Informatik", jeweils in aktueller Ausgabe - empfohlen: Schöning: "Theoretische Informatik kurzgefasst", 5. Auflage, Spektrum, 2008 - Gute Sekundärliteratur: Hopcroft, Motwani, Ullman: "Einführung in die Automatentheorie, Formale Sprachen

	und Komplexitätstheorie", Pearson, 2002 (ein Klassiker...)
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	AC (Aufbaucurriculum / Composition)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture period	Written or oral exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	WiSe	42
Exercises		1	WiSe	14
Präsenzzeit Modul insgesamt				56 h

inf402 - Graph Transformation Systems

Module label	Graph Transformation Systems
Modulkürzel	inf402
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Theoretische Informatik)
Zuständige Personen	Lehrenden, Die im Modul (Prüfungsberechtigt) Lehrenden, Die im Modul (Module responsibility)
Prerequisites	
Skills to be acquired in this module	Modelling of systems, introduction to graph transformation systems, sequential and parallel independence, termination and confluence. **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.
Literaturempfehlungen	Handbook of Graph Grammars and Computing by Graph Transformation, Vol. 1: Foundations, World Scientific, 1997. Vol. 2: Applications, Languages and Tools, World Scientific, 1999. Vol. 3: Concurrency, Parallelism, and Distribution, World Scientific, 1999. H. Ehrig et al.: Fundamentals of Algebraic Graph Transformation. EATCS Monographs of Theoretical Computer Science, Springer, 2006
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	im 2-Jahres-Zyklus
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)

Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	inf401: Theoretische Informatik II

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture period	Written exam or oral exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	SoSe oder WiSe	42
Exercises		1	SoSe oder WiSe	14
Präsenzzeit Modul insgesamt				56 h

inf403 - Cryptology

Module label	Cryptology
Modulkürzel	inf403
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Theoretische Informatik) • Master of Education Programme (Hauptschule and Realschule) Computing Science (Master of Education) > Mastermodule
Zuständige Personen	Lehrenden, Die im Modul (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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
Literaturempfehlungen	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 Cryptography. Anchor, 2000.

Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	Grundveranstaltungen Mathematik und Informatik

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture periode	Written exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		2	WiSe	28
Exercises		2	WiSe	28
Präsenzzeit Modul insgesamt				56 h

inf404 - Petri Nets

Module label	Petri Nets
Modulkürzel	inf404
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Theoretische Informatik)
Zuständige Personen	Lehrenden, Die im Modul (Prüfungsberechtigt) Lehrenden, Die im Modul (Module responsibility)
Prerequisites	
Skills to be acquired in this module	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 poperties • 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. • Unfoldings. • High-level nets.
Literaturempfehlungen	• Best/Wimmel: Skript Petrinetze (Oldenburg, 2015). • Secondary literature: Priese/Wimmel: Petri Netze (Springer-Verlag, 2001).
Links	
Languages of instruction	German, English
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited

Modullevel / module level	AS (Akzentsetzung / Accentuation)			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge				
Examination	Prüfungszeiten		Type of examination	
Final exam of module	At the end of the lecture period		Written or oral exam	
Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		2	WiSe	28
Exercises		2	WiSe	28
Präsenzzeit Modul insgesamt				56 h

inf405 - Algorithmic Graph Theory

Module label	Algorithmic Graph Theory
Modulkürzel	inf405
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Theoretische Informatik)
Zuständige Personen	Lehrenden, Die im Modul (Prüfungsberechtigt) Lehrenden, Die im Modul (Module responsibility)
Prerequisites	
Skills to be acquired in this module	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 G) Approximation Algorithms
Literaturempfehlungen	Jungnickel, Dieter: Graphs, Networks and Algorithms. Springer, Berlin, Heidelberg, 4th edition, 2013. Available as an E-Book in BIS. A detailed bibliography is contained in the lecture notes of this module.
Links	

Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	jährlich
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	Grundveranstaltungen Mathematik und Informatik

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture period	Written exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3	SoSe	42
Exercises		1	SoSe	14
Präsenzzeit Modul insgesamt				56 h

inf407 - Program Verification

Module label	Program Verification
Modulkürzel	inf407
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Bachelor's Programme Computing Science (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Theoretische Informatik)
Zuständige Personen	Olderog, Ernst-Rüdiger (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
Skills to be acquired in this module	Introduction to methods for proving the correctness of sequential, parallel, and distributed programs. **Professional competence** The students: • Describe operational semantics of sequential, parallel, and distributed programs • Know the concepts of partial and total correctness of programs • Establish soundness and completeness of proof systems • Construct input-output specifications of programs • Conduct correctness proofs for programs of different classes with the help of proof rules • Check interference and deadlock freedom of parallel programs • Transform parallel and distributed programs into nondeterministic programs **Methodological competence** The students: • Recognize correctness as an important aspect of programs and informatics systems **Social competence** The students: • Work together in small groups to solve problems • Present their solutions to groups of other students **Self-competence** The students: • Learn persistence in pursuing difficult tasks • Learn precision in specifying problems
Module contents	Program verification is a systematic approach to show the absence of errors in programs. For this purpose desirable behavioural properties of a given program are proven. For instance, a sorting program should only deliver sorted arrays. Partial correctness, termination, and the absence of runtime errors are essential for sequential programs. Additional behavioural properties are of interest for parallel programs: absence of interference, absence of deadlocks, and fair behaviour. The module focuses on the verification of parallel programs. For this purpose classic methods of Hoare's logic are combined with more recent techniques of program transformation. Sequential programs are covered in preparation for this.
Literaturempfehlungen	**essential:** "K.R. Apt, E.-R. Olderog, Programmverifikation, Springer-Verlag, 1994" **Or the extended English version:** "K.R. Apt, F.S. de Boer, E.-R. Olderog, Verfication of Sequential and Concurrent Programs, Third Edition, Springer-Verlag, 2008"

Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	unregelmäßig
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	Theoretische Informatik I und II

Examination	Prüfungszeiten	Type of examination
Final exam of module	At the end of the lecture period	Written exam or oral exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		3		42
Exercises		1		14
Präsenzzeit Modul insgesamt				56 h

inf408 - Algorithms for Software Verification

Module label	Algorithms for Software Verification
Modulkürzel	inf408
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Theoretische Informatik)
Zuständige Personen	Olderog, Ernst-Rüdiger (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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 to 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)
Literaturempfehlungen	• E.M. Clarke, O. Grumberg, and D. Peled: Model Checking. MIT Press, 2000. • F. Nielson, H.R. Nielson, and C. Hankin: Principles of Program Analysis, Springer, 2005 • E.M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, Counterexample-guided abstraction refinement for symbolic model checking, Journal of the ACM 50(5) 752-794 (2003)

Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	unregelmäßig
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	Grundveranstaltungen in Informatik und Mathematik

Examination	Prüfungszeiten	Type of examination
Final exam of module	First week of lecture-free period	Written exam or oral exam

Form of instruction	Comment	SWS	Frequency	Workload of compulsory attendance
Lecture		2	SoSe oder WiSe	28
Exercises		2	SoSe oder WiSe	28
Präsenzzeit Modul insgesamt				56 h

inf409 - Formal Languages

Module label	Formal Languages
Modulkürzel	inf409
Credit points	6.0 KP
Workload	180 h
Verwendbarkeit des Moduls	• Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP) • Master of Education Programme (Gymnasium) Computing Science (Master of Education) > Wahlpflichtmodule (Theoretische Informatik)
Zuständige Personen	Lehrenden, Die im Modul (Prüfungsberechtigt) Lehrenden, Die im Modul (Module responsibility)
Prerequisites	
Skills to be acquired in this module	Introduction to syntactic analysis and compiler construction. **Professional competence** The students: • Know the fundamentals of syntactic analysis and compiler construction • Describe the complexity of fundamental syntactic analysis algortihms • 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)).
Literaturempfehlungen	• J.E. Hopcroft, R. Motwani, J.D. Ullman: Formal Languages and Their Relation to Automata, Addison-Wesley, Boston, 2001. • A.V. Aho, J.D. Ullman: The Theory of Parsing, Translation, and Compiling, Vol. I: Parsing, Prentice-Hall, Englewood-Cliffs, New Jersey, 1972.
Links	
Language of instruction	German
Duration (semesters)	1 Semester
Module frequency	im 2-Jahres-Zyklus
Module capacity	unlimited
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht

Lehr-/Lernform / Teaching/Learning method		V+Ü			
Vorkenntnisse / Previous knowledge		Theoretische Informatik II			
Examination		Prüfungszeiten		Type of examination	
Final exam of module		At the end of the lecture period		Written exam or oral exam	
Form of instruction	Comment	SWS		Frequency	Workload of compulsory attendance
Lecture		2		SoSe	28
Exercises		2		SoSe	28
Präsenzzeit Modul insgesamt					56 h

Abschlussmodul

bam - Bachelor Thesis and Colloquium

Module label	Bachelor Thesis and Colloquium
Modulkürzel	bam
Credit points	15.0 KP
Workload	450 h
Verwendbarkeit des Moduls	• Dual-Subject Bachelor's Programme Computing Science (Bachelor) > Abschlussmodul
Zuständige Personen	Diethelm, Ira (Module responsibility) Lehrenden, Die im Modul (Prüfungsberechtigt)
Prerequisites	
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.
Literaturempfehlungen	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
Modullevel / module level	Abschlussmodul (Abschlussmodul / Conclude)
Modulart / typ of module	Pflicht / Mandatory
Lehr-/Lernform / Teaching/Learning method	SE
Vorkenntnisse / Previous knowledge	

Examination	Prüfungszeiten	Type of examination
Final exam of module	varying	thesis, seminar paper

Form of instruction	Seminar

SWS	2
Frequency	SoSe und WiSe
Workload Präsenzzeit	28 h