fhwedel
UNIVERSITY OF APPLIED SCIENCES

Module Guide
Master-Programme
IT Engineering

Programme Version 20.0

Wedel, den 16. Dezember 2021

Teil I

Module Guide

Kapitel I.1

Module Guide

Index of Modules by Identifier

Index of Modules by Title

I.1.1 Legend to the Module Descriptions

In the following, each module is described in tabular form. The order follows the alphabetical codes.

Preceding the module descriptions there are two directories which support direct access to single descriptions. One directory sorts the descriptions by code, the other by name alphabetically.

The following explanations should make it easier to interpret the entries to the individual fields showing the context in which these entries were made.

Entries for a module

Identifier: Wedel-specific identifier for this module

Name: Name of the module

Course(s): Courses contained in the module with their code and name

Examination: Semester in which the module should be taken assuming a regular course of study

Authority: The strategic tasks of the person in charge of the module comprise of

- synergetic use of the module also in other study programmes
- initiating advancements of the module and its parts
- quality management (e. g. relevance to the programme, work load)

The operative tasks of the module leader comprise in particular of

- coordination of dates for courses and exams.
- initiating and updating the module and course descriptions.
- merging different parts of the exams, collecting the results in tight cooperation with all teachers of the module.
- being the contact person for the students in all questions related to the module .

Curricula: List of all study programmes containing this module

Lessons per Week: Sum of the lessons of all courses contained in this module.

Credits: Sum of credits that can be earned in the courses of this module

Student Workload: The total workload is the number of credits multiplied with 30 hours. The contact time is computed from the lessons per week using the formula 1 lesson = 37.5 minutes. The individual study time is the difference between total workload and contact time.

Prerequesites: Modules, courses and skills that are required at the beginning of the module. If a course is required, the corresponding module is mentioned.

Duration: Number of semesters needed to complete this module

Frequency: indicates how often the module is offered per year
(every semester or every year)

Assessment: List of all types of assessments which are used in some course of the module

Language: In this curriculum all courses are offered in English unconditionally. This applies to courses also used in German study programmes, too. In the other study programmes the courses are offered in German by default. However, some courses are eligible to be taught in English if at least one student of an international partner university is enrolled. Such courses are also listed on a special website in order to allow students of the partner universities to choose a course at home already.

Learning Goals: High level goals referring to the competences to be acquired summarising the individual targets of the course(s)

Applications: Description of exemplary opportunities for practical applications of the module's learning goals

Entries for a course

Course: Name of this course

Lecturer(s): Name(s) of the teachers involved in this course

Recommended Semester: Number of semester in which this course should be attended according to the regular course of studies

Course Type: Compulsory or elected, according to the study programme

Teaching Methods: List of all teaching methods applied.

Credits: Credits to be earned by attendence of this course

Teaching Style: List of all teaching styles applied in this course

Learning Goals: Keywords of the central learning goals of this course

Topics: Structured list of the essential topics covered in this course

References: List of the basic references recommended to the students for reviewing and deepening the essential topics. The list of references actually used may be broader.

I.1.2 Modern Production Methods

M014 Modern Production Methods

Curriculum	Master study programme IT Engineering
Identifier	M014
Name	Modern Production Methods
Course(s)	M014b Laser Engineering M014a Fiber reinforced plastics and hybrids (FRP)
Authority	Dr. Andreas Haase
Curricula	IT Engineering (Master)
Applications	No addictions.
Lessons per Week	4
Credits	5.0
Prerequesites	Basic knowledge of chemistry, materials science and physics, in particular electromagnetic waves and atomic physics.
Duration	1

Learning Goals

In the first part of the module, students learn about modern production processes for processing plastics and elastomers. The students should be able to recognize and name the applied manufacturing processes on the basis of a concrete product. Manufacturing processes are assessed both from a technological and an economic point of view. Students will be able to analyse the requirements of a product for a suitable manufacturing process, select appropriate manufacturing processes and develop new process chains.

In the second part the students get to know the basic physical principles of a laser system. You can name and explain these principles accordingly. The suitability of the laser as a tool in manufacturing is highlighted and distinguished from other manufacturing processes.

The students are able to assign different production requirements to the treated laser systems and are able to make the decision about the selection of a laser system for a concrete production project.

I.1.2.1 Laser Engineering

Course	Laser Engineering
Lecturer(s)	Dominik Miller
Recommended Semester	2
Frequency	annually
Teaching Methods	lecture
Lessons per Week	2
ECTS	2.5
Examination	written or oral examination
Language	english
Teaching Style	blackboard, projector presentation

Learning Goals

After successful completion of this module, students are able to ...

- name and explain the modern production methods for plastics, elastomers and composite materials
- recognize the employed production methods of finished products
- evaluate production methods according to technological and economical aspects
- analyze the demands a product poses for a production method, select corresponding production methods and develop new process chains
- name and explain the physical principles of the laser
- expose the advantages of lasers as production tools and show differences to other methods
- assign different production demands to different laser systems.

Topics

- physical principles
  - electromagnetic radiation
  - elements of atomic physics
  - interaction of light and matter
- laser physics
  - light amplification, population inversion
  - first laser condition
  - dynamics
- laser resonators
  - mirror resonators
  - stability

  - second laser condition
  - resonator modes (longitudinal, transversal)
- laser pulses
  - q-switching
  - fashion coupling
- laser systems
  - distinction of different active media
  - properties of special laser systems
- technological applications

References

- Lasers - Anthony Siegmann, ISBN-0-935702-11-5
- Handbook of Lasers and Optics - Frank Träger (Ed.), Springer Verlag, ISBN-10: 0-387-95579-8, ISBN-13: 978-0-387-95579-7
- Principles of Lasers - Orazio Svelto, Springer Verlag, ISBN 978-1-4419-1301-2

I.1.2.2 Fiber reinforced plastics and hybrids (FRP)

Course	Fiber reinforced plastics and hybrids (FRP)
Lecturer(s)	Hauke Lengsfeld
Recommended Semester	2
Frequency	annually
Teaching Methods	lecture
Lessons per Week	2
ECTS	2.5
Examination	written or oral examination
Language	english
Teaching Style	blackboard, projector presentation

Learning Goals

The students gain ...

- knowledge of FRPs and lightweight materials properties.
- knowledge of manufacturing and assembling technologies.
- basic knowledge of manufacturing engineering and design principles of FRP parts.
- basic skills to assess manufacturing processes from a technical and economical position.

Topics

- basic knowledge of
  - properties of typical fibers and reinforcments
  - semi-finished products: textiles, prepregs
  - manufacturing of fibers and textiles
  - requirements and properties of matrix materials
- engineering with FRPs
  - properties of FRPs, differences to metals
  - engineering design principles with FRPs
  - manufacturing friendly part design
- processing and part manufacturing with FRPs
  - processing and manufacturing technologies
  - processing of semi-finished products (textiles, matrix, prepregs)
  - postprocessing (milling, drilling) of FRP parts
  - assembling technologies of FRP parts
  - recycling
- applications of FRP parts

- space & aeropsace
- automotive
- wind energy & industry

References

I.1.3 Technical Optics

M115 Technical Optics

Curriculum	Master study programme IT Engineering
Identifier	M115
Name	Technical Optics
Course(s)	M115a Technical Optics
Authority	Dr. Andreas Haase
Curricula	IT Engineering (Master)
Applications	The module offers students the opportunity to specialize in an engineering application field. Graduates of this module can work in corresponding companies after their master's degree.
Lessons per Week	4
Credits	5.0
Prerequesites	▪ Knowledge of Physics from the bachelor's programme ▪ Knowledge of Mathematics from the bachelor's programme
Duration	1

Learning Goals

After successful completion of this module, students are able to:

- explain the presented physical concepts and set them in relation to each other
- independently solve problems using the acquired physical concepts and mathematical methods
- critically asses the results and derive conclusions
- design simple lens systems and calculate their optical parameters and aberrations
- explain semiconductor-based light sources and detectors and their role in electronic circuitry
- name different fiber types and their applications in fiber optic communication

I.1.3.1 Technical Optics

Course	Technical Optics
Lecturer(s)	Thomas Pfeuti
Recommended Semester	2
Frequency	annually
Teaching Methods	lecture
Lessons per Week	4
ECTS	5.0
Examination	different types of examinations
Language	english
Teaching Style	blackboard, handout, interactive development and discussion of models, projector presentation, software presentation, student computer exercises

Learning Goals

After successful completion of this module, students are able to ...

- explain the presented physical concepts and set them in relation to each other
- independently solve problems using the acquired physical concepts and mathematical methods
- critically asses the results and derive conclusions
- design simple lens systems, calculate their optical parameters and correct aberrations
- explain semiconductor-based light sources and detectors and their role in electronic circuits
- to name different fiber types and their applications in fiber optic communication, as well as limitations in optical data transmission

Topics

- background:
  - ray optics
  - wave optics
    - interference
    - diffraction
    - resolution limits of optical system
  - beam optics
  - Fourier optics, optical filtering, polarization
- optical imaging
  - optical aberrations

  - the five Seidel aberrations
  - methods for aberration correction
  - development of optical systems
  - specifications of optical systems, dependence of aberrations on optical parameters
  - design programs, tolerancing
- optoelectronics
  - semiconductor photon sources and detectors
- fiber optics
  - fiber types
  - attenuation and dispersion
  - fiber optic communications
- optional advanced topics:
  - modulation, switching, and scanning of light: electrically, acoustically, or optically controlled devices
  - wave interactions in nonlinear materials: frequency conversion
  - holographic

References

- B.E.A. Saleh, M.C.Teich: ”Fundamentals of Photonics”
- G. Litfin: ”Technische Optik”
- F. Pedrotti: ”Introduction to Optics”

I.1.4 Robotics

M018 Robotics

Curriculum	Master study programme IT Engineering
Identifier	M018
Name	Robotics
Course(s)	M018a Robotics
Authority	Prof. Dr. Ulrich Hoffmann
Curricula	Data Science & Artificial Intelligence (Master) IT Engineering (Master)
Applications	The module is reasonably combined with the basic modules ”Introduction to Robotics” and ”Image Processing and Analysis” as well as the module ”Learning & Softcomputing”. It can be used in all technical degree programs.
Lessons per Week	4
Credits	5.0
Prerequesites	Prerequisites are the comprehensive understanding of information technology and software engineering concepts. These are best achieved by a previous studies of computer science or IT engineering with focus on media technology or computer architecture. These studies should have established a bachelor of science in computer science degree. It is assumed that students will be able to work independently in a scientific environment.
Duration	1

Learning Goals

Students earn fundamental competence in selected robot concepts and technlogies.

One focus is to perculate the properties of mobile and autonomous systems. Starting with the basic foundation of robotics topics students wir gain experience in motion and action modelling concepts as well as intelligent learning sensors as basis of autonomous robot behavior.

A showcase implementation within a slef organized group oriented project of one of the theoretically presented concepts enhances the understanding of the concepts at hand.

Students especially achieve a thorough understanding and can categorize and rate practical problems that arise in robot actions guided by visual image processing.

In addition the project leads to an improved presentation style and presentation technique as well as enhanced abilities to freely discuss complex scienticic situations in a team.

I.1.4.1 Robotics

Course	Robotics
Lecturer(s)	Ulrich Hoffmann
Recommended Semester	2
Frequency	annually
Teaching Methods	different types of lectures
Lessons per Week	4
ECTS	5.0
Examination	assessment
Language	english
Teaching Style	

Learning Goals

Students

- have basic knowledge of selected concepts and technologies of robotics.
- thoroughly understand mainly properties of mobile autonomous systems.
- have a deep understanding of the technical foundation of robotics and especially of the concepts of movement and action modeling as well as intelligent learning sensors as basis of autonomous robot behavior.
- are able to realize show case implementations of presented theoretical concepts in a self organized and group oriented project.
- can, based on the concepts presented, independently develop and implement new approaches to solutions and evaluate the result.
- have the competence to understand practical problems that occur when robot actions are guided by visual images.
- are able to convey comprehensibly their scientific results in an appropriate presentation with suitable presentation techniques.
- have the capability to communicate complex scientific facts in a technical discussion in a competent way.

Topics

- Structure and composition of robots
  - kinematics
  - Motion and movers
  - effectors
  - Programming systems
- motion modeling
  - Point to point control

  - Interpolation of trajectories
- Action modeling
- Intelligent sensors
  - Tactile sensors
  - Optical sensors
- Learning robots
- Practical project
  - Independent implementation of a project within the complex subject
  - Experimental exploration of new approaches and ideas that go beyond the lecture content
  - Regular discussion of project results and presentations to all groups

References

- Vukobratovic: Introduction to Robotics, Springer, 1989
- McKerrow: Introduction to Robotics: Introduction to Robotics, Addison Wesley, 1991
- Stienecker: The KUKA Robot Programming Language, self-published, 2011

I.1.5 Industrial Internet of Things

M172 Industrial Internet of Things

Curriculum	Master study programme IT Engineering
Identifier	M172
Name	Industrial Internet of Things
Course(s)	M172a Industrial Internet of Things M172b Industrial Internet of Things Lab
Authority	Prof. Dr.-Ing. Carsten Burmeister
Curricula	IT Engineering (Master)
Applications	Der für diesen Textabschnitt zuständige Mitarbeiter konnte die angeforderten Inhalte leider nicht rechtzeitig liefern.
Lessons per Week	4
Credits	5.0
Prerequesites	Der für diesen Textabschnitt zuständige Mitarbeiter konnte die angeforderten Inhalte leider nicht rechtzeitig liefern.
Duration	1

Learning Goals

Der für diesen Textabschnitt zuständige Mitarbeiter konnte die angeforderten Inhalte leider nicht rechtzeitig liefern.

I.1.5.1 Industrial Internet of Things

Course	Industrial Internet of Things
Lecturer(s)	Carsten Burmeister
Recommended Semester	2
Frequency	annually
Teaching Methods	lecture
Lessons per Week	2
ECTS	3.0
Examination	written or oral examination (+ bonus points)
Language	english
Teaching Style	

Learning Goals

Topics

References

I.1.5.2 Industrial Internet of Things Lab

Course	Industrial Internet of Things Lab
Lecturer(s)	Carsten Burmeister
Recommended Semester	2
Frequency	annually
Teaching Methods	lab
Lessons per Week	2
ECTS	2.0
Examination	None
Language	english
Teaching Style	

Learning Goals

Topics

References

I.1.6 Algorithmics

M003 Algorithmics

Curriculum	Master study programme IT Engineering
Identifier	M003
Name	Algorithmics
Course(s)	M003a Algorithmics
Authority	Prof. Dr. Sebastian Iwanowski
Curricula	Data Science & Artificial Intelligence (Master) IT Engineering (Master)
Applications	The module serves as a starting module. It sets the theoretical fundamentals for a scientific IT oriented study. It covers the knowledge about fundamental algorithms that are necessary for the solution of various application problems.
Lessons per Week	4
Credits	5.0
Prerequesites	▪ Understanding basic mathematical concepts such as definitions, theorems and proofs ▪ Ability of a logically sound formulation ▪ The students must be able to follow proofs from the beginning of this course. ▪ Required is excellent knowledge of the basics of discrete mathematics, specially in number theory and graph theory. ▪ The students must have good programming knowledge and experience in implementing basic algorithms.
Duration	1

Learning Goals

The students know how to evaluate the efficiency of algorithms with theoretically sound methods. For selected application domains, they know how to describe algorithms in detail, show examples and implement them. They are able to solve basic proofs for efficiency and correctness on their own. They can understand even complicated proofs and explain them to other people.

I.1.6.1 Algorithmics

Course	Algorithmics
Lecturer(s)	Sebastian Iwanowski
Recommended Semester	2
Frequency	annually
Teaching Methods	lecture with tutorial, workshop, assignment
Lessons per Week	4
ECTS	5.0
Examination	written or oral examination
Language	english
Teaching Style	blackboard, handout, overhead slide presentation, software presentation

Learning Goals

The students ...

- know the fundamental problems of algorithmics and the classical solving methods.
- are able to analyse the correctness and efficiency of algorithms.
- have detailed knowledge of advanced algorithms for miscellaneous problems in selected application domains.
- know how to implement theoretical results in practical applications.

Topics

- Introduction into formal algorithmics
  - Comparing basic sorting techniques
  - Complexity measures for the analysis of algorithms
  - Lower bound for algorithms using comparisons only
- Advanced searching and sorting
  - Order statistics
  - Searching in sorted arrays
  - Sorting in finite domains
- Solutions for the dictionary problem
  - Hashing and other methods for optimising the average case behaviour
  - (2,3)-trees as example for an optimal worst case behaviour tree
  - Other optimal worst case methods for search trees
  - Optimal binary search trees (Bellman)
- Graph algorithms

    - Minimum spanning trees as motivation for basic algorithms
    - Shortest paths (Dijkstra, Floyd-Warshall, streets)
    - Computation of maximum flows in s/t-networks (Ford-Fulkerson, Edmonds-Karp, Dinic)
    - Computation of graph matchings (bipartite, Edmonds)
- String matching
- Fundamentals of algorithmic geometry
    - Basic problems and the use of Voronoi diagrams for solving them
    - Sweep techniques (including computation of Voronoi diagrams)

References

- deBerg, M., Cheong, O., van Krefeld, M., Overmars, M.:
  Computational Geometry, Algorithms and Applications.
  Springer 2008 (3. edition), ISBN 978-3540779735
- Cormen, T.; Leiserson C.; Rivest, R.; Stein, C.:
  Introduction to Algorithms,
  MIT Press 2001 (2nd ed.)
- Levitin, A.:
  Introduction to the Design and Analysis of Algorithms.
  Addison-Wesley 2006, ISBN 0-321-36413-9
- Mehlhorn, K. / Sanders, P.:
  Algorithms and Data Structures The Basic Toolbox.
  Springer 2008, ISBN 978-3-540-77977-3
- Papadimitriou, C. / Steiglitz, K.:
  Combinatorial Optimization Algorithms and Complexity.
  Dover 1998, ISBN 0-486-40258-4

I.1.7 Cryptography Workshop

M009 Cryptography Workshop

Curriculum	Master study programme IT Engineering
Identifier	M009
Name	Cryptography Workshop
Course(s)	M009a Cryptography Workshop
Authority	Prof. Dr. Gerd Beuster
Curricula	IT Engineering (Master) IT Security (Master)
Applications	For this module, basic knowledge of discrete mathematics is required. The students acquire advanced knowledge about the mathematical basis of cryptography and its practical application. This knowledge can be utilized in all fields where cryptography methods are used.
Lessons per Week	4
Credits	5.0
Prerequesites	Students need the knowledge about discrete mathematics typically acquired in an undergraduate study programs in computer science or a similar field. Students must be familiar with the common Internet protocols. Students must have some basic knowledge in programming.
Duration	1

Learning Goals

In the cryptography workshop, students gain knowledge about the mathematical base of cryptography and its practical application. After completing the course, students are able to use cryptographic methods in the context of secure IT systems, and to evaluate the use of cryptographic methods in existing systems.

This covers both software- and hardware-based cryptography. A focus is put on cryptography used on the Internet and for E-Commerce. The students know how to ensure the confidentiality and integrity of personal data and business data by cryptographic means. Based on real world cryptographic systems, students learned that many side conditions have to be taken into account when implementing and using cryptographic methods.

I.1.7.1 Cryptography Workshop

Course	Cryptography Workshop
Lecturer(s)	Gerd Beuster
Recommended Semester	2
Frequency	annually
Teaching Methods	workshop
Lessons per Week	4
ECTS	5.0
Examination	None
Language	english
Teaching Style	E-Learning, software presentation, student computer exercises

Learning Goals

After completing the module, students are able to ...

- use security tools as an essential building block of modern information and communication systems.
- apply their knowledge of all relevant aspects of data, network and web security.
- asses the application of cryptographic methods, especially for authentication, encryption and integrity preservation.
- assess they algorithmic strengths and weaknesses of cryptographic methods.
- assess and implement cryptographic protocols, especially for authentication in e-commerce.
- consider all side conditions relevant for implementation and application of cryptographic methods.
- assess the quality of random number generators.
- assess the suitability of software and hardware cryptography for a given task.

Topics

- Theory of Cryptography
  - semantic security
  - Unbreakable Encryption and One Time Pad
  - Diffusion and Confusion
- classic cryptography
  - Substitution and Transposition
  - Affine Encryption
  - Rotor Machines
- modern cryptography

    - Stream and Block Ciphers
    - DES and GOST
    - AES
- Block Cipher Modes of Operation
    - ECB, CBC, CTR, AES-GCM
- Random number generators
    - TRNG and PRNG
    - Requirements for CSPRNG
    - PRNG based on mathematical problems
        - Blum Blum Shub
- hashing
    - Hashing Algorithms
        - SHA 2
        - Keccak
    - Message authentication
        - CMAC and HMAC
- asymmetric cryptography
    - Diffie-Hellman
    - RSA
    - elliptic curves
    - Asymmetric Encryption and Digital Signatures
- Practical Cryptography: PGP and SSL
- hardware cryptography
    - trusted computing
    - smart cards
    - Differential Power Analysis

References

- Stallings, William: Cryptography and Network Security : Principles and Practice. 7. Edition. London, UK: Pearson, 2017.
- Ferguson, Niels; Schneier, Bruce; Kohno, Tadayoshi: Cryptography Engineering : Design Principles and Practical Applications. Indianapolis (IN), USA: Wiley Publishing, 2010.
- Menezes, Alfred J.;van Oorschot, Paul C.; Vanstone, Scott A.: Handbook of Applied Cryptography. Boca Raton (FL), USA: CRC Press, 1996.

- Douglas R. Stinson: Cryptography : Theory and Practice. 3. Edition. Boca Raton (FL), USA: CRC Press, 2005.
- Lawrence C. Washington: Elliptic Curves : Number Theory and Cryptography. 2. Edition. Boca Raton (FL), USA: CRC Press, 2008.
- Joshua Davies: Implementing SSL/TLS Using Cryptography and PKI. Indianapolis (IN), USA: Wiley Publishing, 2011.
- Katz, Jonathan; Lindell, Yehuda: Introduction to Modern Cryptography. Boca Raton (FL), USA: CRC Press, 2007.
- Swenson, Christopher: Modern Cryptanalysis : Techniques for Advanced Code Breaking. Indianapolis (IN), USA: Wiley Publishing, 2008.
- Mao, Wenbo: Modern Cryptography: Theory and Practice, Upper Saddle River (NJ), USA: Prentice Hall, 2003.

I.1.8 Security Engineering

M019 Security Engineering

Curriculum	Master study programme IT Engineering
Identifier	M019
Name	Security Engineering
Course(s)	M019a Security Engineering
Authority	Prof. Dr. Gerd Beuster
Curricula	IT Engineering (Master) IT Security (Master)
Applications	The module requires basic knowledge in the fields of computer architecture, operating systems, computer networks, and programming. The skills acquired in this module are applicable to all tasks involving software and security engineering.
Lessons per Week	4
Credits	5.0
Prerequesites	Students must be able to think analytically and to build formal methods. These abilities are typically acquired in an undergraduate study programs in computer science or a similar field. In addition, students must know the general principals of modern computers and operating systems, network technology, and programming.
Duration	1

Learning Goals

After completing the module, the students are able to evaluate the security of existing IT systems and to design and implement new, secure IT systems. This module focuses on the engineering aspects of IT security. When the module is completed, the students know the state of the art in secure software, secure hardware, network security and physical security. The students are able to design systems providing adequate security both for personal and business data.

I.1.8.1 Security Engineering

Course	Security Engineering
Lecturer(s)	Gerd Beuster
Recommended Semester	2
Frequency	annually
Teaching Methods	lecture with tutorial, workshop, assignment
Lessons per Week	4
ECTS	5.0
Examination	written or oral examination
Language	english
Teaching Style	E-Learning, interactive development and discussion of models, software presentation, student computer exercises

Learning Goals

After completing the module, students are able to ...

- apply the basic concepts of IT Security.
- define and check security requirements for software.
- develop and evaluate secure software.
- assess and evaluate the security of hardware components.
- evaluate the security of computer networks.
- design secure computer networks.

Topics

- Basic Concepts of IT Security
- threat modeling
- Threats in Practice
- security modeling
- Security Administration and Physical Security
- Operating System Security and Access Rights
- security protocols
- Methods for Developing Secure Software
- Typical Attacks on Software Systems
- Distributed Systems / Network Security
- Secure Hardware

References

- Allen, Julia H.; Barnum, Sean; Ellison, Robert J.; McGraw, Gary; Mead, Nancy R.: Software Security Engineering : A Guide for Project Managers. Bosten (MA), USA: Addison Wesley, 2008.

- Anderson, Ross J.: Security Engineering : A Guide to Building Dependable Distributed Systems. 2. Edition. Hoboken (NJ), USA: Wiley & Sons, 2008.

- Graves, Michael W.: Digital Archaeology : The Art and Science of Digital Forensics. Bosten (MA), USA: Addison Wesley, 2013.

- Pfleeger, Charls P.;Pfleeger, Shari Lawrence: Security in Computing. 4. Edition. München: Prentice Hall, 2012.

- Shimeall, Timothy J.; Spring, Jonathan M.: Introduction to Information Security : A Strategic-based Approach. Amsterdam, NL: Elsevier Syngress, 2013.

- Stallings, William: Computer Security : Principles and Practice. 2. Edition. München: Pearson, 2012.

- Watson, David; Jones, Andrew: Digital Forensics Processing and Procedures. Amsterdam, NL: Elsevier Syngress, 2013.

- Wilhelm, Thomas: Professional Penetration Testing : Creating and Operating a Formal Hacking Lab. 2. Edition. Amsterdam, NL: Elsevier, 2013.

I.1.9 Seminar IT Engineering

M041 Seminar IT Engineering

Curriculum	Master study programme IT Engineering
Identifier	M041
Name	Seminar IT Engineering
Course(s)	M041a Seminar IT Engineering
Authority	jeweiliger Dozent
Curricula	IT Engineering (Master)
Applications	for project and master thesis
Lessons per Week	2
Credits	5.0
Prerequesites	none besides the skills achieved in a Bachelor's programme
Duration	1

Learning Goals

In the seminar, students should become familiar with scientific work.

I.1.9.1 Seminar IT Engineering

Course	Seminar IT Engineering
Lecturer(s)	jeweiliger Dozent
Recommended Semester	1
Frequency	every semester
Teaching Methods	seminar
Lessons per Week	2
ECTS	5.0
Examination	written documentation (if necessary presentation)
Language	english
Teaching Style	blackboard, handout, overhead slide presentation, projector presentation, software presentation

Learning Goals

In the seminar, students should learn to prepare and present a complex subject. A written paper is also required, which is to be similar in style to a thesis, but much smaller in scope. This will familiarize students with the formalities of a thesis.

Topics

The topics may come from all areas of engineering sciences or computer science.

References

must be found by the student. For some topics there are suggestions from the supervising lecturer which should be extended.

I.1.10 Project IT Engineering

M040 Project IT Engineering

Curriculum	Master study programme IT Engineering
Identifier	M040
Name	Project IT Engineering
Course(s)	M040a IT Engineering Project
Authority	Prof. Dr.-Ing. Carsten Burmeister
Curricula	IT Engineering (Master)
Applications	The projects requires the students to combine their software development skills and their engineering skills.
Lessons per Week	2
Credits	5.0
Prerequesites	Software development skills and basic engineering knowledge is required.
Duration	1

Learning Goals

After completing this module the students can combine their technology knowledge with their learned software skills to develop and implement solution strategies within a practical project.

I.1.10.1 IT Engineering Project

Course	IT Engineering Project
Lecturer(s)	verschiedene Dozenten
Recommended Semester	2
Frequency	every semester
Teaching Methods	project
Lessons per Week	2
ECTS	5.0
Examination	written documentation (if necessary presentation)
Language	english
Teaching Style	blackboard, projector presentation, student computer exercises

Learning Goals

After completing the project, the students have developed the ability to aquire knowledge on their own. They can work on a project by describing the problem, modelling a system, develop a solution strategy and implement this.

Topics

depending on the student-selected topic

References

Depending on project topic

I.1.11 Security Management

M049 Security Management

Curriculum	Master study programme IT Engineering
Identifier	M049
Name	Security Management
Course(s)	M049a Security Management
Authority	Prof. Dr. Gerd Beuster
Curricula	Business Information Technology/IT Management (Master) IT Engineering (Master) IT Security (Master)
Applications	The module is part of the specialisation "Information Management" within the Master in Business Administration Program. The module does not require specific knowledge, but general analytical thinking and modelling skills are required. The knowledge acquired in the module can be used both in the area of security management and in other management areas, in particular in quality management.
Lessons per Week	4
Credits	5.0
Prerequesites	Students need the analytical thinking and modelling skills acquired in a bachelor's degree in computer science or similar studies.
Duration	1

Learning Goals

In the Security Management module, students learn to evaluate and design IT security in the context of corporate strategies. Students learn to understand security as a holistic concept that not only has software, but also hardware as well as administrative and physical aspects. After completing the module, they will be familiar with the legal and private sector standards for safety evaluation and certification. You can create and implement security concepts and policies. They are familiar with the basic concepts of data protection in the national and international context. Students are taught the ability to take on management tasks in the field of IT security and to work as IT security managers. They are in a position to identify goods worth protecting in a company and to develop and implement the administrative measures necessary for protection. The students know the interfaces to and overlaps with other areas of management, in particular IT management and change management.

I.1.11.1 Security Management

Course	Security Management
Lecturer(s)	Gerd Beuster
Recommended Semester	1
Frequency	annually
Teaching Methods	lecture with tutorial, workshop, assignment
Lessons per Week	4
ECTS	5.0
Examination	written or oral examination
Language	english
Teaching Style	E-Learning, interactive development and discussion of models, software presentation

Learning Goals

In the Security Management module, students learn to evaluate and design IT security in the context of corporate strategies. Students are taught the ability to take on management tasks in the field of IT security and to work as IT security managers.

They acquire the ...

ability to identify and model threats.
Ability to evaluate risks.
Ability to assess the adequacy of security measures and to design appropriate security measures.
Knowledge of the relevant standards and certification schemes in the field of IT security
Ability to ensure IT security in interaction with organizational and physical security requirements and measures
Knowledge of the connections between safety and quality management

Topics

- Introduction to IT Security Management
- Corporate security as an economic factor
- Attackers and targets
- Management of security-critical IT projects
- Basic IT-Protection
- Evaluation and certification schemes in IT security
- Data protection
- Safety training courses
- Physical safety
- Security audits and revision control
- Safety management and quality management

References

I.1.12 Dynamical Systems

M168 Dynamical Systems

Curriculum	Master study programme IT Engineering
Identifier	M168
Name	Dynamical Systems
Course(s)	M168a Dynamical Systems
Authority	Prof. Dr.-Ing. Carsten Burmeister
Curricula	IT Engineering (Master)
Applications	The module is to be combined with othe modules from the foeld of engineering, mathematics and applied computer science.
Lessons per Week	4
Credits	5.0
Prerequesites	The students should have basic knowledge from the field of signals and linear systems.
Duration	1

Learning Goals

After completing this module the students have basic knowledge on describing, modelling and solving dynamical systems. They can desribe multi-dimensional systems by means of a state space representaion. They can solve linear systems exactly and non-linear systems approximatly.

I.1.12.1 Dynamical Systems

Course	Dynamical Systems
Lecturer(s)	Carsten Burmeister
Recommended Semester	1
Frequency	annually
Teaching Methods	lecture with tutorial, workshop, assignment
Lessons per Week	4
ECTS	5.0
Examination	written or oral examination
Language	english
Teaching Style	blackboard, handout, interactive development and discussion of models, projector presentation, software presentation, student computer exercises

Learning Goals

The students ...

- have a basic knowledge on modelling dynamical systems with differential equations,
- are able to transfer systems into state space representation,
- are able to solve systems approximatly, calculate fixed points and determine their stability.

Topics

- One-dimensional systems
  - Differential Equation models
  - Geometric models
  - Fixed Points and stability
  - Approximate solutions using MATLAB
  - Birfurcations
- Linear multi-dimensional systems
  - State space representation
  - solving linear systems
  - solving linear systems in MATLAB
- Non-linear multi-dimensional systems
  - Grafische Modellierung in der Phasen-Ebene
  - Fix-Punkt Untersuchungen durch Linearisierung

References

Steven Strogatz, Nonlineal Dynamics and Chaos, Perseus Books Publishing, 1994.

I.1.13 Embedded Systems Workshop

M038 Embedded Systems Workshop

Curriculum	Master study programme IT Engineering
Identifier	M038
Name	Embedded Systems Workshop
Course(s)	M038a Embedded Systems Workshop
Authority	Dipl.-Ing. (FH) Timm Bostelmann
Curricula	IT Engineering (Master)
Applications	This module fits in with other modules covering hardware and software engineering.
Lessons per Week	6
Credits	5.0
Prerequesites	Knowledge of electronics engineering and system programming
Duration	1

Learning Goals

The students know how to engineer a prototype of an embedded system based on a functional specification. Furthermore they are able to perform methodical tests and create a technical documentation.The students are able to utilize data-sheets to read up on complex components like micro-controllers and embedded sensors. They can engineer embedded hardware and handle the specific demands of embedded software development.

I.1.13.1 Embedded Systems Workshop

Course	Embedded Systems Workshop
Lecturer(s)	Timm Bostelmann
Recommended Semester	1
Frequency	annually
Teaching Methods	workshop
Lessons per Week	6
ECTS	5.0
Examination	acceptance test
Language	english
Teaching Style	blackboard, handout, projector presentation, software presentation, student computer exercises

Learning Goals

The students are able to ...

- engineer a prototype of an embedded system based on a functional specification.
- understand and utilize data-sheets.
- read up on complex components like micro-controllers and embedded sensors.
- engineer analogue and digital interface hardware for an embedded system (depending on the workshop topic).
- engineer embedded software.
- create a technical documentation.

Topics

- Introduction to embedded systems.
  - Embedded system engineering
  - Embedded hardware engineering
  - Embedded software engineering
- Introduction to the laboratory equipment.
- Guided engineering of a simple embedded system.
- Engineering, testing and documentation of an embedded system prototype.

References

I.1.14 Medical Engineering

M059 Medical Engineering

Curriculum	Master study programme IT Engineering
Identifier	M059
Name	Medical Engineering
Course(s)	M059a Medical Engineering
Authority	PD Dr. Dennis Säring
Curricula	IT Engineering (Master)
Applications	not specified
Lessons per Week	4
Credits	5.0
Prerequesites	Basics in Physics Basics in Programming
Duration	1

Learning Goals

Medical Datasets and Information

- Where can Computer Science help in medicine?
- Patient records and clinical documentation
- Coding and Diagnosis Related Groups

Medical Imaging

- Working Principle And Algorithms For Image Aquisition (EEG, US, CT, MRI, Nuclear Medicine)

Medical Image Processing

- Working princples of image pre-processing
- Basics about image segmentation and image analysis

I.1.14.1 Medical Engineering

Course	Medical Engineering
Lecturer(s)	Dennis Säring
Recommended Semester	1
Frequency	annually
Teaching Methods	lecture
Lessons per Week	4
ECTS	5.0
Examination	written or oral examination
Language	english
Teaching Style	E-Learning

Learning Goals

Medical Datasets and Information

- Where can Computer Science help in medicine?
- Patient records and clinical documentation
- Coding and Diagnosis Related Groups

Medical Imaging

- Working Principle And Algorithms For Image Aquisition (EEG, US, CT, MRI, Nuclear Medicine)

Medical Image Processing

- Working princples of image pre-processing
- Basics about image segmentation and image analysis

Topics

Students ...

- shall understand the importance of the fields of engineering and computer science for medicine.
- shall understand that designing and approving medical devices.
- will learn to understand the different physical working principles and main algorithms used for different medical imaging devices such as Ultrasonography, Electroencephalography, X-ray Computed Tomography and Magnetic Resonance Imaging
- will learn some basics in medical image processing

References

not specified

I.1.15 Distributed Systems

M035 Distributed Systems

Curriculum	Master study programme IT Engineering
Identifier	M035
Name	Distributed Systems
Course(s)	M035a Distributed Systems M035b Tutorial: Distributed Systems
Authority	Prof. Dr. Ulrich Hoffmann
Curricula	Business Information Technology/IT Management (Master) Computer Science (Master) IT Engineering (Master) IT Security (Master)
Applications	The module can well be combined with modules "Functional Programming" and "Current Developments in Computer Science" as well as with the "Seminar Master".
Lessons per Week	4
Credits	5.0
Prerequesites	The practical exercises assume advanced programming abilities. In addition the module assume solid knowledge of internet architecture and structure as well as basic knowledge of enterprise workflow process organization.
Duration	1

Learning Goals

Students gain extended knowledge of technical aspects of distributed systems as well as their area of applications in commercial contexts. They experience and discuss technological inherent problems of distributed systems and thus have the ability to address the challenges, such as questions of IT security and encrypted communications, of distributed system and to cope with them. They know the architecture and major algorithms in distributed systems as well as processes in development and administration that lead to successful distributed products. They are able to program distributed systems in different programming paradigms.

I.1.15.1 Distributed Systems

Course	Distributed Systems
Lecturer(s)	Ulrich Hoffmann
Recommended Semester	1
Frequency	annually
Teaching Methods	lecture
Lessons per Week	2
ECTS	3.0
Examination	written or oral examination
Language	english
Teaching Style	

Learning Goals

The students gain ...

- thorough understanding of principles of distributed applications.
- knowledge in mastering base technologies and current software tools for distributes systems.
- knowledge of state of the are in different application areas such as service mediation and e-commerce.
- knowledge of IT security concerns in distributed systems as well as of encrypted communications.
- knowledge of basic algorithms in distributed systems.
- precise knowledge of current web service architectures.
- practical skills to realize a project.
- distributed programming skills in different paradigms.

Topics

- practical examples
- general requirements of distributed systems
- the client server relation and resulting questions
- communications in distributed systems
- naming services
- techniques for concurrency
- remote calls
- alternative paradigms (actor concept, \ldots)
- synchronization of data and processes
- coordination methods

- replication techniques
- WEB services with SOAP and REST
- fault tolerance concepts
- security in distributed systems
- programming with threads
- communication via sockets, structure of clients and servers
- remote procedure call / remote method invocation
- using naming services
- programming WEB services (SOAP, server / client, WSDL, data binding)
- distributed programming with alternate concepts
- programming synchronization algorithms
- programming distributed election algorithms
- programming of REST based services and clients
- fault tolerant programming in distributed systems

References

- ARMSTRONG, Joe:
  Programming Erlang.
  Pragmatic Programmers, 2007
- ODERSKY, Martin; SPOON, Lex; VENNERS, Bill:
  Programming in Scala.
  Artima Press, Mountain View, 2008
- COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim:
  Distributed Systems, Concepts and Design.
  Addison-Wesley, 2011, ISBN 0-1321-4301-1
- TANENBAUM, Andrew; VAN STEEN, Marten:
  Distributed Systems, Priciples and Paradigms.
  Prentice Hall, 2006, ISBN 0-1323-9227-5

I.1.15.2 Tutorial: Distributed Systems

Course	Tutorial: Distributed Systems
Lecturer(s)	Ulrich Hoffmann
Recommended Semester	1
Frequency	annually
Teaching Methods	tutorial/lab/business game
Lessons per Week	2
ECTS	2.0
Examination	acceptance test
Language	english
Teaching Style	

Learning Goals

The students ...

- have the ability to operate typical software systems (middleware) in the area of distributed systems and use them to solve problems.
- are accustomed to problems that occur in reality and are able to overcome these.
- have some hands on experience with IT security issues.
- know how to use encryption in distributed setups.
- have deep knowledge of the specific properties of distributed systems by practical experience. They can categorize and evaluate these properties.

Topics

Lecture accompanying practical exercises in programming distributed systems and their algorithms in different programming paradigms.

References

- c., f. lecture
- numerous online ressources

I.1.16 Master Thesis

M060 Master Thesis

Curriculum	Master study programme IT Engineering
Identifier	M060
Name	Master Thesis
Course(s)	M060a Master Thesis
Authority	jeweiliger Dozent
Curricula	IT Engineering (Master)
Applications	None.
Lessons per Week	0
Credits	28.0
Prerequesites	A prerequisite for the Master's thesis is the material from the previous two semesters, in particular the courses that are related to the topic of the thesis.
Duration	1

Learning Goals

In the Master's thesis, the students show that they are able to work independently and goal-oriented on complex tasks with a scientifically methodical approach. They are able to locate problems in a larger context, to network the technical connections and to present the findings convincingly and argumentatively.

I.1.16.1 Master Thesis

Course	Master Thesis
Lecturer(s)	jeweiliger Dozent
Recommended Semester	3
Frequency	every semester
Teaching Methods	thesis
Lessons per Week	0
ECTS	28.0
Examination	written documentation (if necessary presentation)
Language	english
Teaching Style	none

Learning Goals

The students are able to ...

- to develop complex tasks independently.
- locate problems in a larger context.
- use scientific methods for problem solving.
- convincingly present results.

Topics

topic-dependent

References

topic-dependent

I.1.17 Master Colloquium

M061 Master Colloquium

Curriculum	Master study programme IT Engineering
Identifier	M061
Name	Master Colloquium
Course(s)	M061a Colloquium
Authority	jeweiliger Dozent
Curricula	IT Engineering (Master)
Applications	None
Lessons per Week	0
Credits	2.0
Prerequesites	The prerequisite for admission to the colloquium is a Master’s thesis that has been assessed as at least ”sufficient”.
Duration	1

Learning Goals

The students present their work results convincingly to the examination board. They master the instrument of free speech, argue conclusively and provide evidence.
In a subsequent interdisciplinary oral examination, they defend their work results and prove to be familiar with problems in the discussion.

I.1.17.1 Colloquium

Course	Colloquium
Lecturer(s)	jeweiliger Dozent
Recommended Semester	3
Frequency	every semester
Teaching Methods	colloquium
Lessons per Week	0
ECTS	2.0
Examination	colloquium
Language	english
Teaching Style	blackboard, interactive development and discussion of models, overhead slide presentation, software presentation

Learning Goals

The students ...

- have the ability to concentrate on an intensively worked subject.
- strengthen the competence to conduct a technical discussion about problem solving and its quality.
- have strong communication and presentation skills.

Topics

- Lecture on the topic of the Master's thesis as well as on the chosen procedure and the results
- Discussion of the quality of the chosen solution
- Questions and discussion on the subject of the Master's thesis and related fields

References

topic-dependent

I.1.18 Additional Records (30 ECTS)

AdRe Additional Records (30 ECTS)

Curriculum	Master study programme IT Engineering
Identifier	AdRe
Name	Additional Records (30 ECTS)
Course(s)	AdRe1 Discrete Mathematics AdRe7 Application of Artificial Intelligence AdRe3 Fundamental Programming Structures Lab Course AdRe6 Signals and Systems AdRe4 Programming Structures Project AdRe0 Programming Qualifyer AdRe2 Fundamental Programming Structures AdRe5 Advanced Programming Features
Authority	Prof. Dr. Sebastian Iwanowski
Curricula	IT Engineering (Master)
Applications	for all master modules
Lessons per Week	27
Credits	30.0
Prerequesites	Ability to think abstractly and logically, at least basic programming knowledge
Duration	2

Learning Goals

The prescribed modules from the bachelor's level serve to acquire the theoretical and practical knowledge that we expect from a student with a bachelor's degree in IT engineering or an equivalent course.

I.1.18.1 Discrete Mathematics

Course	Discrete Mathematics
Lecturer(s)	Sebastian Iwanowski
Recommended Semester	1
Frequency	annually
Teaching Methods	lecture
Lessons per Week	4
ECTS	5.0
Examination	written examination
Language	english
Teaching Style	blackboard, overhead slide presentation, projector presentation, software presentation, tutorials

Learning Goals

At the end of the course, the students have the following competences:

- Knowledge of basic mathematical terms and concepts (definition, proposition, proof) and ability to distinguish them.
- Understanding the fundamentals and formalization of logical reasoning.
- Understanding elementary logic and set theory and the mutual connection between these disciplines.
- Based on those disciplines, understanding the concept of relations and functions.
- Ability to apply elementary proof principles such as mathematical induction in different contexts.
- Knowledge of the basic propositions of elementary number theory, group and field theory, combinatorics and graph theory, application on examples on their own.

Topics

- Logic
  - Introduction
  - Propositional logic
  - Predicate logic
- Set theory
  - Basic terms and concepts
  - Relations
  - Functions
  - Boolean algebras
- Proof concepts

  - Glossary of mathematical structures
  - Mathematical induction
  - Other proof strategies
- Number theory
  - Divisibility
  - Division with remainder
  - Prime numbers
  - Modular arithmetic
- Algebraic structures
  - groups
  - fields
- Combinatorics
  - Enumeration formulae for finite sets
  - Permutations
- Graph theory
  - Terminology and representation
  - Paths problems in graphs
  - Trees
  - Planar graphs
  - Graph colouring

References

- Norman L. Biggs: *Discrete Mathematics*, Oxford University Press 2002 (2. edition), ISBN 0-19-850717-8
- Neville Dean: *The Essence of Discrete Mathematics*, Prentice Hall 1997, ISBN 0-1334-5943-8
- Susanna S. Epp: *Discrete Mathematics with Applications*, Brooks/Cole 1995 (2. edition), ISBN 0-534-94446-9
- Jiri Matousek / Jaroslav Nesetril: *An Invitation to Discrete Mathematics*, Oxford University Press 2008 (2. edition), ISBN 0-1985-7042-2
- Kenneth H. Rosen: *Discrete Mathematics and its Applications*, McGraw-Hill 2003, ISBN 0-07-242434-6

I.1.18.2 Application of Artificial Intelligence

Course	Application of Artificial Intelligence
Lecturer(s)	Sebastian Iwanowski
Recommended Semester	1
Frequency	annually
Teaching Methods	lecture with tutorial, workshop, assignment
Lessons per Week	4
ECTS	5.0
Examination	written examination
Language	english
Teaching Style	

Learning Goals

At the end of the course, the students have the following competences:

- Knowledge and interest in the basic goals of Artificial Intelligence.
- Knowledge of the basic technologies of artificial intelligence.
- Ability to apply elementary techniques of artificial intelligence in implementations.
- Knowledge of various complex application examples.

Topics

- Introduction
  - Definition and objectives of the AI
  - Overview of the basic technologies of the KI
  - Selection of application examples
- Basic Technologies
  - Knowledge-based systems with the subdivisions rule-based, model-based and case-based reasoning
  - Machine Learning
  - Search strategies
  - Swarm intelligence
  - Fundamentals of semantic networks
- Applications
  - Traffic information and navigation
  - Logistic problems
  - Technical diagnosis
  - Image recognition

    - Games

References

- Marco Dorigo / Thomas Stützle:
  Ant Colony Optimization,
  MIT Press 2004, ISBN 0-262-04219-3
- Goodfellow, Ian, Yoshua Bengio und Aaron Courville: Deep Learning. http://www.deeplearningbook.org. MIT Press, 2016. ISBN: 978-0-262-03561-3
- Stuart Russell / Peter Norvig:
  Artificial Intelligence: A modern approach,
  Pearson Studium 2010 (3. edition), ISBN 978-0-13-207148-2
- Liyang Yu : A Developer's Guide to the Semantic Web , Springer 2011, ISBN 978-3-642-15969-5

I.1.18.3 Fundamental Programming Structures Lab Course

Course	Fundamental Programming Structures Lab Course
Lecturer(s)	Cordula Eichhorn Marcus Riemer
Recommended Semester	1
Frequency	every semester
Teaching Methods	lecture
Lessons per Week	2
ECTS	2.0
Examination	acceptance test
Language	english
Teaching Style	online content, projector presentation, software presentation, student computer exercises, tutorials

Learning Goals

In the lab course, the students acquire the ...

- Ability to apply the basic concepts of object-oriented programming languages in practice and to implement them in Java.
- Ability to build simple dynamic data structures in the context of an object-oriented programming language and ability to apply basic algorithms to these data structures.
- Ability to use a current, widely used development environment.
- Ability to implement complete software systems on a small scale based on a written task.
- Ability to develop software in a small team.
- Ability to identify suitable test cases for quality assurance.
- Knowledge of the basic rules for designing user-friendly interfaces and software.

Topics

- Using already acquired basic programming skills of the students to introduce them to programming with Java and the development environment.
- Application of the basic concepts of object-oriented programming presented in the lecture by solving written assignments in small teams.
- Testing and presenting of the cleanly structured solution.

References

Barry Burd: Beginning Programming with Java (for dummies), 2017 (5th edition) ISBN: 978-1-11-923553-8

I.1.18.4 Signals and Systems

Course	Signals and Systems
Lecturer(s)	Carsten Burmeister
Recommended Semester	2
Frequency	annually
Teaching Methods	lecture with tutorial, workshop, assignment
Lessons per Week	4
ECTS	5.0
Examination	different types of examinations
Language	english
Teaching Style	blackboard, handout, projector presentation, software presentation, student computer exercises

Learning Goals

The students ...

- know time-continuous and time-discrete integral transformations and their properties.
- can use integral transformations for spectral analysis, for solving linear differential equation systems and for general analysis of communication and control systems.

Topics

- Description and analysis of continuous signals and systems in the time domain
  - Signals and signal types
  - Systems and System Properties
  - Impulse response and convolution
- Description and analysis of continuous signals and systems in the frequency domain
  - Fourier analysis
  - Fourier series
  - Fourier transform
- Description and analysis of continuous signals and systems in the image area
  - laplace transformation
  - system stability
- Description and analysis of time-discrete signals and systems
  - Sampling / sampling theorem
  - Digital Signal Processing
  - z-transformation
  - Discrete Fourier Transform and FFT

References

Meyer, M.: Signalverarbeitung. Vieweg und Teubner 2011, 6. Auflage

Oppenheim, A., Willsky, S.: Signals and Systems, 2nd Edition. Pearson 1996, 2. Auflage

Chaparro, L.: Signals and Systems using MATLAB. Academic Press 2018, 3. Auflage

I.1.18.5 Programming Structures Project

Course	Programming Structures Project
Lecturer(s)	Nils van Kan
Recommended Semester	1
Frequency	annually
Teaching Methods	lecture
Lessons per Week	4
ECTS	5.0
Examination	acceptance test
Language	english
Teaching Style	projector presentation, student computer exercises

Learning Goals

With this project the students acquire the ...

- Ability to use the basic concepts of object-oriented programming languages and their implementation in Java in a medium sized software project.
- Ability to use current development environments.
- Ability to independently structure and realize complete software systems of medium complexity based on a problem-oriented task definition.
- Ability to develop and document software.
- Ability to apply the basic rules of user friendly interface design.

Topics

- Development of a complete software system of medium complexity in Java based on a problem-oriented task.
- Structuring and modularization of the project.
- Independent design of suitable data models.
- User-oriented design of the user interface.
- Testing of the developed software and documentation of the tests.
- Creating a documentation of the program and its development and creating a user manual.

References

I.1.18.6 Programming Qualifyer

Course	Programming Qualifyer
Lecturer(s)	Sebastian Iwanowski
Recommended Semester	1
Frequency	every semester
Teaching Methods	tutorial/lab/business game
Lessons per Week	1
ECTS	.0
Examination	acceptance test
Language	english
Teaching Style	

Learning Goals

The students should be enabled to pass the other programming courses. Above all, they should be able to solve new tasks by thinking on their own and not by copying other templates.

Topics

- Data types and variables
- Control structures: Branch, loop, recursion
- Procedures and functions
- Modular reasoning
- Very simple algorithms

References

I.1.18.7 Fundamental Programming Structures

Course	Fundamental Programming Structures
Lecturer(s)	Cordula Eichhorn Marcus Riemer
Recommended Semester	1
Frequency	annually
Teaching Methods	lecture
Lessons per Week	4
ECTS	3.0
Examination	written examination
Language	english
Teaching Style	blackboard, handout, online content, projector presentation

Learning Goals

The students ...

- identify the basic concepts of object-oriented programming and compare them with the concepts of procedural programming.
- develop software based on the core concepts of object-oriented programming.
- compile the basic language elements (data types, instructions, realization of object-oriented concepts) of Java and select from them to develop Java programs of medium complexity.
- use a modern development environment to support software development and present the associated functionalities and procedures.
- design simple dynamic data structures in the context of an object-oriented programming language.
- explain basic algorithms that work on the mediated data structures.
- design appropriate modularization for programs of medium complexity by using suitable elements of the Java programming language and define corresponding interfaces between the modules.
- name the basic rules for the user-friendly design of programs and use them to design the user interfaces of programs with limited functionality both structurally and functionally appropriately.
- know the fundamental classes and their operations, which are neccessary to implement operations on files and data streams

Topics

- Basic concept of the Java programming language
  - Basic properties of the language
  - Basic structure of Java programs
  - Execution of Java programs

- Presentation of the development environment used (NetBeans)
- Basic program elements
  - Primitive Data Types in Java
  - Variables, Assignments, Scopes of Validity
  - Operators and Expressions
  - Instructions
- Reference data types
  - Arrays
  - Classes
- Static Methods
- Basic Classes
  - String
  - StringBuilder
  - Enum
- Basic concepts of object orientation
  - Classes and Instances with Attributes and Methods
  - Visibility, Packages
  - Constructors
  - Inheritance and overwriting
  - Dynamic binding, polymorphism
  - Object-oriented realization of recursive dynamic data structures (lists)
  - Generic Types
  - Abstract Classes and Interfaces - Declaration and Use
  - Realization of graphical user interfaces
  - Handling runtime errors
  - Classes for the realization of file operations

References

Barry Burd: Beginning Programming with Java (for dummies), 2017 (5th edition) ISBN: 978-1-11-923553-8

I.1.18.8 Advanced Programming Features

Course	Advanced Programming Features
Lecturer(s)	nicht benannt
Recommended Semester	2
Frequency	annually
Teaching Methods	tutorial/lab/business game
Lessons per Week	4
ECTS	5.0
Examination	acceptance test
Language	english
Teaching Style	blackboard, handout, overhead slide presentation, projector presentation, software presentation, student computer exercises

Learning Goals

The students...

- apply advanced concepts of object-oriented programming languages such as parametric polymorphism and multiple inheritance using Java as an example.
- name and explain the basics of generic abstract data types and their correspondence with concepts of object-oriented languages (interfaces, abstract classes, concrete classes, polymorphism).
- develop and design generic abstract data types and their implementation.
- use prefabricated container structures such as the Java Collections Framework in a systematic way and select abstract data types and implementations that are appropriate to the problem.
- name and explain motivations, basics and problems of concurrent programming.
- use primitives of concurrent programming in Java (creating threads, thread communication, synchronization, and so on)

Topics

Applied abstract data types

- Abstract data types and their implementions using Java Collections as an example
- Concepts of type systems including the example of Java Generics
- Concepts of memory allocation, especially continuous and chained memory management

Concurrent programming

- Concurrent algorithms (potential, challenges, formulation)
- Fundamentals of multi-threading using Java as an example (threads, monitors, waitsets, volatile variables)
- Frameworks for the formulation of concurrent processing (e.g. Java Streams)

References

- Gosling, James; Joy, Bill; Steele, Guy; Bracha, Gilad; Buckley, Alex: The Java Language Specification, Java SE 8 Edition, 2015
- Bloch, Joshua: Effective Java: A Programming Language Guide, 3rd Edition, Addison-Wesley, 2017
- Urma, Raoul-Gabriel; Fusco, Mario; Mycroft, Alan: Java 8 in Action: Lambdas, Streams, and Functional-Style Programming, Manning Publications, 2014
- Sedgewick, Robert; Wayne, Kevin: Algorithms, 4th Edition, Addison-Wesley, 2011
- Goetz, Brian: Java Concurrency in Practice, Addison-Wesley, 2006
- Herlihy, Maurice; Shavit, Nir: The Art of Multiprocessor Programming, Revided Reprint, Morgan Kaufmann, 2012

Dokumenttyp	Modulhandbuch
Abschlusstyp	Master
Studiengangname	IT Engineering
Ordnungsnummer	20.0
Setzdatum	16. Dezember 2021
git	ja
git-commit	99b702a1 (lokale Änderungen vorhanden)