Hochschule Karlsruhe

Faculty for Computer Science and Business Information Systems

Module manual

Course of studies Computer Science (Master), ER 8

Summer semester 2025

Module Computer Science (Master), ER 8

Module Interaction Design	
Internal number	INFM110MI
Coordinator	Prof. Thomas Hinz
Scope	7.0 ECTS points, 6.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	Students learn theoretical knowledge of designing interactive systems and put their knowledge into practice in constructing prototypes. With knowledge of design principles and concepts they are able to solve design problems. They gain practical experiences on innovative natural user interfaces and interaction in public space and can develop new forms of human-machine interfaces.
Exams	Individual exams
Lecture Design of Interactions	
Internal number	INFM111MI
Lecturer	Prof. Thomas Hinz
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	Students acquire the ability to conceptualise and design future-oriented interactive systems, taking into account societal, technological and social aspects. They can apply design futuring methods for the systematic development of future scenarios, analyse complex socio-technical systems and their interactions, create participative design processes for the development of sustainable future solutions, develop prototypes and future scenarios that can be experienced, and critically reflect on the social implications of technological innovations.

Recommended reading	Script, Literature: - B. Groß, E. Mandir: Zukünfte gestalten: Spekulation. Kritik. Innovation. Mit »Design Futuring« Zukunftsszenarien strategisch erkunden, entwerfen und verhandeln.. 2022, ISBN-13: 978-3874399586 - M. Leube: The Future Designer: Anthropology Meets Innovation in Search of Sustainable Design (Routledge Earthscan), 2024, ISBN-13 : 978-1032735535 - A. Dunne, F. Raby: Speculative Everything: Design, Fiction, and Social Dreaming, 2013, ISBN-13:978-0262019842 - T. Pyczak: Tell me!: Wie Sie mit Storytelling überzeugen. Zahlreiche Praxisbeispiele für alle, die erfolgreich sein wollen in Beruf, PR und Marketing., 2023, ISBN-13:978-3836291545 - R. Klanten, L. Feireiss, "A Touch of Code: Interactive Installations and Experiences", 2011, ISBN-13 978-3899553314 - J. Sauter, S. Jaschko, J. Ängeslevä, ART+COM: "Medien, Räume und Installationen", 2011, ISBN-13 978-3899553864 - H.-D. Hellige, "Mensch-Computer-Interface: Zur Geschichte und Zukunft der Computerbedienung", 2008, ISBN-13 978-3899425642 - J. Schenk, G. Rigole, "Mensch-Maschine-Kommunikation: Grundlagen von sprach- und bildbasierten Benutzerschnittstellen", 2010, ISBN-13 978-3642054563 - R. Dorau, "Emotionales Interaktionsdesign Gesten und Mimik interaktiver Systeme", 2011, ISBN-13 978-3642031007
Exams	Verbal Exam/Concept 20/1 Min./Semester (graded)
Comments	Participation at tuition, in class group work and discussion.

Lecture Interaction Design Exercise

Internal number	INFM112MI
Lecturer	Prof. Thomas Hinz
Scope	4.0 ECTS points, 4.0 Contact hours 120 Stunden gesamt, davon 60 Stunden Kontaktstudium.
Type/mode	Exercise
Language of instruction	German
Content	The participants apply their theoretical understanding of the conception and design of interactive systems. They design and develop prototypes of interactive systems. The students experiment with innovative forms of human-computer interfaces and their possibilities. They are capable of presenting their results convincingly using multimodal tools, as well as justify them in a methodical and theoretically grounded manner.
Recommended reading	- Lecture materials
Exams	Exercise 1 Semester (not graded)
Comments	- Teamworking - Experiments on design and prototyping - Presentation and discussion of the results

Module Machine learning	
Internal number	INFM110ML
Coordinator	Prof. Dr. Dennis Janka
Scope	7.0 ECTS points, 6.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	
Exams	Individual exams
Lecture Machine learning	
Internal number	INFM111ML
Lecturers	Prof. Dr. Christine Preisach Prof. Dr. Dennis Janka
Scope	4.0 ECTS points, 4.0 Contact hours 120 Stunden gesamt, davon 60 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	
Recommended reading	
Exams	Written/verbal Exam 120/20 Min. (graded)
Comments	
Lecture Machine learning Exercise	
Internal number	INFM112ML
Lecturers	Prof. Dr. Dennis Janka Prof. Dr. Christine Preisach
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Exercise
Language of instruction	German
Content	
Recommended reading	
Exams	Exercise 1 Semester (graded)
Comments	

Module Software Architectures	
Internal number	INFM110SE
Coordinator	Prof. Dr. Thomas Fuchß
Scope	7.0 ECTS points, 6.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	On successful completion of the module, students will be able to independently design, develop and critically evaluate the architecture of large-scale software systems. They will be able to decompose software into components and connectors, select and evaluate appropriate frameworks, and logically organise functionalities and workflows. Students will be able to plan and design parallel and concurrent processes and integrate them into architectures tailored to different application domains. In addition, they will have developed their technical, social and personal skills, as well as their communication and self-management skills, and will be able to apply them effectively in the context of large-scale software projects.
Exams	Individual exams
Lecture Software Architectures	
Internal number	INFM111SE
Lecturer	Prof. Dr. Thomas Fuchß
Scope	2.0 ECTS points, 2.0 Contact hours 60 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German

Content	The "Software Architectures" course provides advanced knowledge and skills in the development and analysis of modern software architectures. In the first part of the lecture, relevant process models are reviewed, and approaches for agile architecture development are introduced. Students explore and compare various view models, deepen their understanding of modeling techniques for components and connectors, and expand their knowledge of design methods. The course emphasizes the use of scenarios for describing and evaluating non-functional requirements. The second part focuses on fundamental architectural patterns, such as layered architecture, hexagonal architecture, onion architecture, blackboard, pipes-and-filters, and event-driven architecture. Students learn to identify, understand, and apply these patterns while exploring their role in typical middleware concepts. The lecture demonstrates how these architectural templates provide pathways to organizing flexible and evolutionary systems. The third part addresses architecture at the module level. Using practical scenarios and examples, the application of patterns for the business layer is presented and analyzed in context. The course aims to equip students with a comprehensive understanding of designing and evaluating modular software architectures.
Recommended reading	- Avgeriou, P; et. al (editors): Relating Software Requirements and Architectures. Springer, 2011. - Clements, P.; Bass, L. and Kazman, R.: Software Architecture in Practice, 2. ed. Addison-Wesley, 2003. - Fowler, M.: Patterns of Enterprise Application Architecture. Addison-Wesley, 2003. - Goll, J. und Dausmann, M.:Architektur- und Entwurfsmuster der Softwaretechnik. Springer Vieweg, 2013. - Gorton, Ian: Essential Software Architecture, 2. ed. Springer, 2011. - Larman, Craig: Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3. ed. Prentice Hall, 2004. - Lilienthal, Carola: Sustainable software architecture: analyze and reduce technical debt. dpunkt.verlag, 2019. - Buschmann, Frank: A System of Patterns (POSA V.1). John Wiley & Sons. 1996 - Schmidt, Douglas C.: Patterns for Concurrent and Networked Objects (POSA V.2). John Wiley & Sons, 2000. - Sommerville, Ian: Software Engineering, 9. Auflage. Pearson Studium, 2012. - Vogel, O.; Arnold, I.; Chughtai, A. and Kehrer, T.: Software Architecture: A Comprehensive Framework and Guide for Practitioners. Springer, 2011. - Vogel, O.; et. al: Software-Architektur: Grundlagen – Konzepte – Praxis, 2. Auflage. Spektrum, 2009.
Exams	Written/verbal Exam 60/20 Min. (graded)
Comments	The lecture will take the form of seminars with exercises and presentations.
Lecture Parallel Programming	
Internal number	INFM112SE

Lecturer	Dipl.-Ing. Christian Meder
Scope	2.0 ECTS points, 2.0 Contact hours 60 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Laboratory Course
Language of instruction	German
Content	The "Parallel Programming Lab" provides practical knowledge of concurrent programming and its application across various modern technologies and systems. In the first part of the course, students revisit and deepen their understanding of the fundamentals of concurrent programming. Through hands-on exercises using Java and the Java Development Kit (JDK), they learn how to efficiently design concurrent processes and address challenges such as race conditions and deadlocks. In the second part, the course explores practical applications of concurrent programming through concise introductions to various technologies and platforms, including: - Concurrency in operating systems, e.g., Linux - Concurrency on mobile platforms, e.g., Android - Map/Reduce and the Hadoop ecosystem - Concurrency in web applications - Concurrency in modern programming languages like Go and Rust - Reactive Extensions (Rx) and their use cases - Cloud, cluster, and distributed systems - Microservice architectures - Deep learning and concurrency By combining theoretical insights with practical exercises, students gain the skills to understand, implement, and advance concurrent programming in diverse technical contexts.
Recommended reading	- Brian Goetz, "Java Concurrency in Practice" - Online Tutorials The literature for the second part changes every semester.
Exams	Exercise 1 Semester (not graded)
Comments	Passing the lab will be assessed by completing the exercises independently. In the second part of the lab, various new methods and concepts will be presented and discussed.
Lecture Software Architectures Laboratory	
Internal number	INFM113SE
Lecturer	Prof. Dr. Carsten Sinz
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Laboratory Course
Language of instruction	German

Content	In this lab, the methods and concepts from the Software Architectures lecture are practically applied and tested. Using a larger software project as a case study, the entire development process — from design and implementation to testing and quality assurance — is carried out in a team. Design techniques and architectural patterns from the lecture should be applied, and software development frameworks should be evaluated and selected accordingly. The project also includes implementation in a programming language chosen by the team, along with quality assurance, allowing for a critical assessment of the applied techniques. Finally, by creating both a developer handbook and user manual, students will learn how to document larger projects effectively.
Recommended reading	
Exams	Exercise 1 Semester (not graded)
Comments	

Module Smart Interaction	
Internal number	INFM120MI
Coordinator	Prof. Dr. Matthias Wölfel
Scope	7.0 ECTS points, 6.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	
Exams	Individual exams
Lecture Smart Interaction	
Internal number	INFM121MI
Lecturer	Prof. Dr. Matthias Wölfel
Scope	3.0 ECTS points, 3.0 Contact hours 90 Stunden gesamt, davon 45 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	Nowadays machines are already capable of communicating with human beings in a "natural" fashion through the existence of capabilities to understand natural language, recognise hand writing, and for interpreting gestures. However, they are also capable of extending human perception through augmenting situations with additional knowledge ("augmented reality"), i.e. the depiction of information is contextualised according to the situation as perceived by the machine. Examples of this are smartphones and tablets (voice-control, face recognition, "goggle", music recognition), vehicles (driver assistance systems), video game consoles (movement interpretation), but also in work-related contexts (surgery, human-robot-cooperation). The lecture covers the foundations of voice- and gesture recognition, the sensing and recognition of objects in the environment, as well as information presentation. Sample applications (e.g. using the Kinect sensor) allow students to gain deeper understanding of the covered material. Topics include: - system performance of perception-based interaction - sensor systems for the recognition of the environment (sound, video, 3d, touch, acceleration and rotation) - Recognition (object recognition in video and 3d, speech- and behaviour recognition) - interaction models (augmented reality, situation graphs)
Recommended reading	
Exams	Verbal Exam/Concept 20/1 Min./Semester (graded)
Comments	

Lecture Smart Interaction Exercise	
Internal number	INFM122MI
Lecturer	Prof. Dr. Matthias Wölfel
Scope	4.0 ECTS points, 3.0 Contact hours 120 Stunden gesamt, davon 45 Stunden Kontaktstudium.
Type/mode	Exercise
Language of instruction	German
Content	
Recommended reading	
Exams	Exercise 1 Semester (not graded)
Comments	

Module Data Science	
Internal number	INFM120ML
Coordinator	Prof. Dr. Reimar Hofmann
Scope	7.0 ECTS points, 6.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	At the end of this module, students will have understood the theoretical and mathematical foundations of machine learning and data analysis. They will be able to assess the suitability of different methods for specific situations, interpret phenomena they observe conclusively and, if necessary, derive ideas for improving the selected approaches. The skills taught in the module are advantageous for participation in the Artificial Intelligence module.
Exams	Written/verbal Exam 120/20 Min. (graded)
Lecture Data Science	
Internal number	INFM121ML.a
Lecturer	Prof. Dr. Reimar Hofmann
Scope	2.0 ECTS points, 2.0 Contact hours 60 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	- Scale types (nominal, ordinal, interval, ratio), conversion of scale types (one-hot-coding etc.), standardization - Explorative data analysis, differentiation between direct and indirect dependencies, data visualization - Statistical principles of machine learning, maximum likelihood approach, bias and variance (overfitting) as sources of learning error - Cost functions for numerical regression and classification - Criteria foe data quality, dealing with quality deficiencies (e.g. missing values, outliers) - Dealing with more complex data types (record data, heterogeneous data, bag of words), data transformations, feature engineering - Dimension reduction: heuristic, manual, etc. - Volume reduction: sampling etc.
Recommended reading	
Exams	Module exam
Comments	
Lecture Optimization	

Internal number	INFM121ML.b
Lecturer	Prof. Dr.-Ing. Astrid Laubenheimer
Scope	2.0 ECTS points, 2.0 Contact hours 60 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	
Recommended reading	
Exams	Module exam
Comments	
Lecture Optimization Exercise	
Internal number	INFM122ML
Lecturer	Prof. Dr.-Ing. Astrid Laubenheimer
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Exercise
Language of instruction	German
Content	
Recommended reading	
Exams	Exercise 1 Semester (graded)
Comments	

Module Programming language concepts	
Internal number	INFM120SE
Coordinator	Prof. Dr. Martin Sulzmann
Scope	7.0 ECTS points, 6.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	Programming languages are the most basic tool of programmers. This course covers the basics of the theory of programming languages, its applications to widely used languages today, and trends in the field that are likely to help define the programming languages of tomorrow. After this course, students will have an understanding of a wide range of programming language concepts and will be able to quickly adapt to emerging new programming languages. A mix of written as well as practical exercises will help students to get an understanding of different styles of programming in several programming languages. Content. Various styles of programming with a focus on functional programming: data types and pattern matching higher-order functions polymorphic types Different flavors of polymorphism: subtyping (structual and nominal) generics, overloading Program correctness: static verification methods testing methods such as QuickCheck There is no final exam. Evaluation is based on quizzes, midterms and several smaller projects that are carried out during the semester.
Exams	Individual exams
Lecture Programming paradigms	
Internal number	INFM121SE
Lecturer	Prof. Dr. Martin Sulzmann
Scope	3.0 ECTS points, 3.0 Contact hours 90 Stunden gesamt, davon 45 Stunden Kontaktstudium.
Type/mode	Lecture

Language of instruction	German
Content	Programming languages are the most basic tool of programmers. This course covers the basics of the theory of programming languages, its applications to widely used languages today, and trends in the field that are likely to help define the programming languages of tomorrow. After this course, students will have an understanding of a wide range of programming language concepts and will be able to quickly adapt to emerging new programming languages. A mix of written as well as practical exercises will help students to get an understanding of different styles of programming in several programming languages. Content: Various styles of programming with a focus on functional programming: - data types and pattern matching - higher-order functions - polymorphic types Different flavors of polymorphism: - subtyping (structual and nominal) - generics, - overloading Program correctness: - static verification methods - testing methods such as QuickCheck There is no final exam. Evaluation is based on quizzes, midterms and several smaller projects that are carried out during the semester.
Recommended reading	- Lecture notes and slides - Exercises - Textbook: Concepts in Programming Languages von John C. Mitchell
Exams	Exercise 1 Semester (graded)
Comments	Prerequisites: Experience in a programming language. Mix of lectures (2/3) and practical exercies (1/3)
Lecture Program Optimization Laboratory	
Internal number	INFM122E
Lecturer	Prof. Dr. Christian Pape
Scope	4.0 ECTS points, 3.0 Contact hours 120 Stunden gesamt, davon 45 Stunden Kontaktstudium.
Type/mode	Exercise
Language of instruction	German

Content	The students learn to correctly implement complex algorithms using a system-level procedural or object-oriented programming language such as C or C++. They optimize these algorithms in terms of execution speed using various paradigms such as hybridization or cache optimization techniques. The participants in the course measure the execution times and empirically check the asymptotic runtime of the implemented procedures. To do this, they create various test scenarios in order to compare and evaluate the algorithms based on them. The students document and discuss the results. They analyze their optimizations and explain the results. The computer science problems to be addressed vary. Typical content is: - Ray tracing - Fast multiplication of polynomials and numbers - Graphical geometric algorithms - Graph algorithms
Recommended reading	Detailled description of the assigments and source code framework.
Exams	Exercise 1 Semester (not graded)
Comments	The solutions are presented and discussed with the supervisor during the exercises.

Module Game Design	
Internal number	INFM210MI
Coordinator	Prof. Daniel Schwarz
Scope	7.0 ECTS points, 5.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	
Exams	Individual exams
Lecture Game Design	
Internal number	INFM211MI
Lecturer	Prof. Daniel Schwarz
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	
Recommended reading	
Exams	Verbal Exam/Concept 20/1 Min./Semester (graded)
Comments	
Lecture Game Design Exercise	
Internal number	INFM212MI
Lecturer	Prof. Daniel Schwarz
Scope	4.0 ECTS points, 3.0 Contact hours 120 Stunden gesamt, davon 45 Stunden Kontaktstudium.
Type/mode	Exercise
Language of instruction	German
Content	
Recommended reading	
Exams	Exercise 1 Semester (not graded)
Comments	

Module Artificial Intelligence	
Internal number	INFM210ML
Coordinator	Prof. Dr. Patrick Baier
Scope	7.0 ECTS points, 6.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	Machine learning
Pre-requisites according to the examination regulations	none
Competences	
Exams	Individual exams
Lecture Artificial Intelligence	
Internal number	INFM211ML
Lecturers	Prof. Dr. Patrick Baier Prof. Dr. Jannik Strötgen
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German

Content	This lecture covers current developments and recent research findings in the field of Artificial Intelligence, with a particular focus on Deep Learning. After completing the lecture, students are able to assess the fundamentals of neural networks and evaluate current research methods in the domains of Computer Vision and Natural Language Processing. Students will be capable of evaluating basic architectures such as Convolutional Neural Networks and Recurrent Neural Networks, as well as analyzing advanced architectures like Transformer models. Furthermore, students are able to assess appropriate architectures for application areas in Computer Vision and Natural Language Processing, comprehend scientific literature such as recent publications, and analyze specific aspects. Additionally, they will understand current methods from subfields such as Continual Learning. The lecture serves as a theoretical foundation for the overall module “Artificial Intelligence” and complements the practical course “AI Lab.” Practical exercises are therefore not part of this lecture. Group presentations are an option for a deeper exploration of specific topics. The lecture covers the following topics: - Review of the fundamentals of neural networks and Deep Learning - Various architectures of CNNs - Application examples in Computer Vision, such as Object Detection and Instance Segmentation - Models for processing sequential data, such as RNNs and (Bi-)LSTMs - Application examples in Natural Language Processing, such as Machine Translation and Information Extraction - Language models and word embeddings - Attention mechanism and Transformer models - Large language models (LLMs) - Core cross-domain methods like Transfer Learning - Introductions to specialized areas such as Explainable AI, Continual Learning (e.g., for improving LLMs), or Diffusion Models Students' understanding of the fundamental concepts and architectures of Deep Learning, as well as their ability to evaluate current research findings, will be assessed through an exam. Answering the questions will require both individual formulations and performing calculations based on examples from the application domains. In addition, students have the option to demonstrate their ability to evaluate and explain specific topics in detail through a voluntary group presentation on a specialized topic. A presentation that is convincing due to thorough analysis and clear explanation will result in an improvement of the exam grade by one grade level.
Recommended reading	
Exams	Written/verbal Exam 60/20 Min. (graded)

Comments	
Lecture Artificial Intelligence Exercise	
Internal number	INFM212ML
Lecturer	Dr. Patrick Baier
Scope	4.0 ECTS points, 4.0 Contact hours 120 Stunden gesamt, davon 60 Stunden Kontaktstudium.
Type/mode	Exercise
Language of instruction	German
Content	This lab implements the theoretical foundations from the lecture into practical tasks. For this, tasks from the following three domains are tackled: * Computer Vision * Natural Language Processing * Reinforcement Learning
Recommended reading	
Exams	Exercise 1 Semester (graded)
Comments	Requirements: - Basic knowledge in Python - Basic knowledge in Machine Learning

Module Theory of efficient algorithms	
Internal number	INFM210SE
Coordinator	Prof. Dr. Heiko Körner
Scope	7.0 ECTS points, 5.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	Students will be able to design efficient algorithms in theory and practice. They will be able to prove the correctness of various graph-theoretical problems with exact logical conclusions. They are able to analyse the time complexity of algorithms and use suitable analysis techniques. They will also be able to independently implement modelling and simulation methods for the computer-aided design of process sequences and parallelise various iteration methods on modern high-performance computers.
Exams	Written/verbal Exam 120/20 Min. (graded)
Lecture Graph Algorithms	
Internal number	INFM211SE.a
Lecturer	Prof. Dr. Heiko Körner
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	The course teaches basic algorithms on graphs and shows how to analyse their correctness and time complexity. After a brief introduction to graph theory, we will first introduce search methods such as breadth-first and depth-first search. Further algorithms deal with the recognition of strongly connected components, topological sorting and finding shortest paths. Efficient tests for the circularity of graphs are also discussed. The lecture enables participants to independently develop further algorithms, apply them in a provably safe manner and evaluate their usefulness.
Recommended reading	- Discussion at the blackboard - Lecture notes - Sample solutions for all exercises - T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein: Introduction to Algorithms, 4th edition. MIT Press, 2022.
Exams	Module exam
Comments	Classical lecture. Several exercises deepen the field of study and are discussed in the classroom if desired.
Lecture SAT Solving	
Internal number	INFM211SE.a

Lecturer	Prof. Dr. Carsten Sinz
Scope	2.0 ECTS points, 2.0 Contact hours 60 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	SAT solving is one of the most important general methods for solving difficult (often NP-complete) combinatorial problems. These occur in practice in a variety of applications, e.g.: - Planning and scheduling problems in supply chains - Configuration of complex, multi-variant products, e.g. cars, lorries, aircraft - Testing (verification) of hardware and software - Creation of match schedules, e.g. soccer leagues This module teaches students the theoretical and mainly practical aspects of SAT-Solving. It deals with: - Fundamentals, historical development - Coding, e.g. cardinality constraints - Phase transitions in random problems - Local search (GSAT, WalkSAT, ..., ProbSAT) - Resolution, Davis-Putnam algorithm, DPLL-algorithm, look-ahead algorithm - Efficient implementations, data structures - Heuristics in the DPLL-algorithm - CDCL-algorithm, closed-loop learning, implication graphs - Restarts and heuristics in the CDCL-algorithm - Preprocessing, inprocessing - Generation of proofs and their verification - Parallel SAT-Solving (Guiding Paths, Portfolios, Cube-and-Conquer) - Advanced applications: Bounded model checking, planning, satisfiability-modulo-theories Usually industrial users are also integrated.
Recommended reading	
Exams	Module exam
Comments	

Lecture SAT Solving Exercise

Internal number	INFM212SE
Lecturer	Prof. Dr. Carsten Sinz
Scope	2.0 ECTS points, 1.0 Contact hours 60 Stunden gesamt, davon 15 Stunden Kontaktstudium.
Type/mode	Exercise
Language of instruction	German

Content	In this exercise, methods of the lecture Practical SAT Solving are tested on the basis of practical problems and SAT-solvers are used to solve combinatorial problems.
Recommended reading	
Exams	Exercise 1 Semester (not graded)
Comments	

Module Special chapters Media Informatics	
Internal number	INFM220MI
Coordinator	Prof. Dr. Matthias Wölfel
Scope	7.0 ECTS points, 5.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	
Exams	Individual exams
Lecture Special chapters Media Informatics	
Internal number	INFM221MI
Lecturers	Prof. Daniel Schwarz Prof. Thomas Hinz Dr. Tim Schlippe Prof. Dr. Matthias Wölfel
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	
Recommended reading	
Exams	Verbal Exam/Concept 20/1 Min./Semester (graded)
Comments	
Lecture Special chapters Media Informatics Exercise	
Internal number	INFM222MI
Lecturers	Prof. Thomas Hinz Prof. Daniel Schwarz Prof. Dr. Matthias Wölfel
Scope	4.0 ECTS points, 3.0 Contact hours 120 Stunden gesamt, davon 45 Stunden Kontaktstudium.
Type/mode	Exercise
Language of instruction	German
Content	
Recommended reading	
Exams	Exercise 1 Semester (not graded)
Comments	

Module Special chapters KI	
Internal number	INFM220ML
Coordinator	Prof. Dr. Patrick Baier
Scope	7.0 ECTS points, 5.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	
Exams	Written/verbal Exam 120/20 Min. (graded)
Lecture Special chapters KI 1	
Internal number	INFM221ML
Lecturer	Dr. Patrick Baier
Scope	4.0 ECTS points, 3.0 Contact hours 120 Stunden gesamt, davon 45 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	
Recommended reading	
Exams	Written/verbal Exam 60/20 Min. (graded)
Comments	
Lecture Explainable AI	
Internal number	INFM221ML.a
Lecturer	Prof. Dr. Jannik Strötgen
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German

Content	This lecture deals with a sub-area of artificial intelligence, Explainable AI (XAI). XAI methods fulfill a critical function in the modern AI landscape by bridging the gap between human users and complex AI systems. After attending the lecture, students will be able to describe the diverse goals of XAI and classify and evaluate numerous methods from the field of explainable artificial intelligence using the correct XAI terminology. Students will also be able to evaluate the underlying ideas, architectures and implications for dedicated XAI methods, as well as understand scientific literature such as current publications and highlight important aspects. Students will also be able to apply methods and use libraries through practical exercises. The following topics will be covered in this lecture with methods with different (i) scope (local vs. global), (ii) result (e.g. feature relevance, surrogate models), (iii) functioning (e.g. perturbations, examples), (iv) format (e.g. textual, visual) and (v) stage (post-hoc vs. ante-hoc): - Motivation and goals of XAI - XAI terminology and XAI taxonomy - Methods from the area of feature importance explanations (e.g. feature attribution vs. feature selection) - Methods from the area of removal-based explanations (e.g. LIME, RISE) - Shapley values (and SHAP) - Methods from the area of concept-based explanations (e.g. concept bottleneck models) - Methods for interpreting neural networks (e.g. attention-based methods) - Potential dangers of XAI methods, for example, with regard to "fairwashing" and "manipulation". At the end of the lecture, an oral exam is held to check the extent to which students have understood basic XAI concepts and to which extent they can evaluate current research results. Answering the questions requires both the correct use of the terminology learned and the ability to analyze and, in some cases, evaluate different methods. In addition to group exercises to understand individual methods that are then explained to other students through short presentations, students also have the opportunity to demonstrate that they can evaluate and explain individual topics in detail with a larger, voluntary group presentation on a special topic. A presentation that is convincing due to thorough analysis and clear explanation will result in an improvement of the exam grade by one grade level.
Recommended reading	- lecture slides - Molnar, C. (2024). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable (2nd ed.). christophm.github.io/interpretable-ml-book/
Exams	Module exam
Comments	
Lecture Special chapters KI 2	
Internal number	INFM222ML

Lecturer	Dr. Patrick Baier
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	
Recommended reading	
Exams	Written/verbal Exam 60/20 Min. (graded)
Comments	
Lecture Summer School Advanced Topics in AI	
Internal number	INFM223ML
Lecturer	Dr. Patrick Baier
Scope	4.0 ECTS points, 3.0 Contact hours 120 Stunden gesamt, davon 45 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	
Recommended reading	
Exams	Module exam
Comments	
Lecture Winter School Advanced Topics in AI	
Internal number	INFM224ML
Lecturer	Dr. Patrick Baier
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	
Recommended reading	
Exams	Module exam
Comments	

Module Special chapters on software engineering	
Internal number	INFM220SE
Coordinator	Prof. Dr. Dirk Hoffmann
Scope	7.0 ECTS points, 5.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	
Exams	Individual exams
Lecture Coding Theory	
Internal number	INFM221SE
Lecturer	Prof. Dr. Dirk Hoffmann
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	This course teaches foundations about codes and codings as well as basic algorithms for source coding, channel coding and line coding. The lecture focuses on the following topics: information and coding theory, data compression, error detecting and correcting codes, limits of data transmission.
Recommended reading	Slides, blackboard, exercise sheets
Exams	Written/verbal Exam 60/20 Min./Semester (graded)
Comments	Lecture
Lecture Modeling and Simulation	
Internal number	INFM222SE
Lecturer	Prof. Dr. Britta Nestler
Scope	2.0 ECTS points, 2.0 Contact hours 60 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	
Recommended reading	
Exams	Written/verbal Exam 60/20 Min./Semester (graded)
Comments	
Lecture Modeling and Simulation Exercise	
Internal number	INFM222SE

Lecturer	Prof. Dr. Britta Nestler
Scope	2.0 ECTS points, 1.0 Contact hours 60 Stunden gesamt, davon 15 Stunden Kontaktstudium.
Type/mode	Exercise
Language of instruction	German
Content	
Recommended reading	
Exams	Exercise 1 Semester (not graded)
Comments	

Module Mobile and Distributed Systems	
Internal number	INFM230SE
Coordinator	Prof. Dr. Oliver Waldhorst
Scope	7.0 ECTS points, 6.0 Contact hours
Placement	All semesters
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	The module provides in-depth knowledge and practical skills in the field of mobile and distributed systems. Students develop an in-depth understanding of the underlying technologies and architectures as well as the competence to design, evaluate and apply modern solutions for complex, real-world problems. - In lecture Mobile Systems, students acquire the ability to analyze mobile communication systems, identify their challenges and apply suitable technologies such as MANETs, DTNs or mobile TCP. - The lecture Distributed Systems enables students to understand and evaluate current technologies such as cloud computing, IoT and DLT and to apply them in a global context. - The Distributed Systems Lab offers practical experience in the development of distributed systems, in particular the migration of monolithic architectures to microservices. Students use modern technologies such as Kubernetes, Docker and Istio. After completing the module, students will be able to critically evaluate technological concepts of mobile and distributed systems, design their integration and implement innovative solutions in a professional and research-oriented context.
Exams	Written/verbal Exam 120/20 Min. (graded)
Lecture Mobile Systems	
Internal number	INFM231.a
Lecturer	Prof. Dr. Oliver Waldhorst
Scope	2.0 ECTS points, 2.0 Contact hours 60 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German

Content	In this course, students acquire a basic understanding of the challenges and technologies in the field of mobile communication systems. After completing the lecture, they will be able to explain basic principles of wireless and mobile communication and analyze and evaluate specific technologies such as mobility models, mobile ad hoc networks (MANETs), delay tolerant networks (DTNs) and mobile TCP. In addition, they are able to identify problems in the field of mobile systems and apply suitable solutions in practical scenarios. The lecture covers the following topics: · Fundamentals of mobile systems: Challenges posed by mobility, wireless communication and networks. · Mobility models: simulation and analysis of individual and group-based mobility. · Mobile ad hoc networks (MANETs): Self-organizing networks, routing protocols and use cases. · Delay-tolerant networks (DTNs): Communication with intermittent connectivity and "store-carry-forward" mechanisms. · Mobile TCP: Adaptation and optimization of the Transmission Control Protocol for mobile and wireless networks. The lecture is taught in a flipped classroom format. Students prepare for the live sessions independently with the help of lecture slides and explanatory videos. In the classroom sessions, content is deepened through case studies and exercises. Online tests are available for self-assessment, offering students feedback and the opportunity to consolidate what they have learned. The examination consists of a 60-minute written exam, which is part of the module exam "Mobile and Distributed Systems". The total workload is 60 hours, which is divided into 20 hours of attendance time in the live sessions, 20 hours of asynchronous learning with lecture slides and videos and 20 hours for exam preparation and follow-up work.
Recommended reading	- Slide collection and explanatory videos in the ILIAS system - James Kurose, Keith Ross: Computer Networking - A Top-Down Approach, 8th edition, Pearson, 2021 (Chapter 7). - Martin Sauter, Grundkurs Mobile Kommunikationssysteme, 8th edition, 2022 (available as an e-book via the KIT library) - Further information in ILIAS and in the lecture
Exams	Module exam
Comments	The lecture will take the form of seminars with exercises.

Lecture Distributed Systems

Internal number	INFM231SE
Lecturer	Prof. Dr. Christian Zirpins
Scope	2.0 ECTS points, 2.0 Contact hours 60 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture

Language of instruction	German
Content	Building on an assumed understanding of the basic principles and paradigms of distributed systems, this Master's course deals with case studies of current application areas. The selection of covered contents varies. On the one hand, practically significant (industry-relevant) areas are considered. On the other hand, current trends from research and development are addressed. Currently, the course focuses on the topic of internet computing. In this course, students gain a comprehensive understanding of the infrastructure and technologies that underpin today's Internet computing. They will develop a solid understanding of distributed system architectures and web technologies that are essential for navigating the complex landscape of modern IT environments. By exploring current paradigms such as cloud computing and the Internet of Things (IoT), students will be able to evaluate and utilize these technologies in a variety of professional contexts. In addition, by exploring emerging technologies such as distributed ledger technologies and fog computing, students will be prepared to develop innovative contributions in the field of Internet computing. The course aims to enhance their analytical skills and enable them to critically evaluate the integration and potential of Internet-based technologies in shaping individual, organizational and societal practices. In addition, students will enhance their research and inquiry skills through engagement with a variety of learning resources, including examples, further reading and comprehension questions in the accompanying textbook.
Recommended reading	Lecture Notes: - Sunyaev, Ali. 2020. Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-34957-8. An extensive bibliography and specific recommendations for further reading will be presented during the course.
Exams	Module exam
Comments	Independent work units concern the follow-up of the course content and the exam preparation.

Lecture Distributed Systems Laboratory

Internal number	INFM232SE
Lecturer	Prof. Dr. Christian Zirpins
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Laboratory Course
Language of instruction	German

Content	The lab provides practical insights into the construction of distributed information systems. Current paradigms are taken up and extended principles are dealt with in the context of realistic use cases. The specific tasks are based on current topics in industrial research and development. It therefore varies from semester to semester. The practical implementation is carried out using modern industry-relevant platforms and frameworks. Currently, the lab covers a project to migrate a monolithic information system following the microservice architectural style. It utilizes technologies like UML and Domain Driven Design, REST-based Microservices with Spring/Spring-Boot, a Microservice-Platforms with Docker and Kubernetes as well as Service Meshes based on Istio. Through participation in this lab class, students will gain hands-on experience in designing, developing, and deploying distributed information systems, particularly through the lens of converting monolithic architectures into microservices. They will become proficient in using a suite of modern, industry-standard tools and technologies, including UML for modeling, Domain-Driven Design for structuring systems, Spring and Spring Boot for creating REST-based microservices, and Docker and Kubernetes for containerization and orchestration. Furthermore, the application of Istio for managing service meshes will equip students with the skills necessary for optimizing the communication and operation of microservices in complex systems. This practical experience will prepare students for current and future challenges in industrial research and development.
Recommended reading	- S. Newman, "Microservices - Konzeption und Design", mitp, 2015 - E. Wolf, Das Microservices-Praxisbuch: Grundlagen, Konzepte und Rezepte, dpunkt, 2018, - B. Rumpe , Modellierung mit UML, Xpert.press, 2011 - V. Vernon, Domain-Driven Design kompakt, dpunkt, 2017 - E. Wolf, 2016, Microservices - Grundlagen flexibler Softwarearchitekturen, dpunkt, 2016 - E. Wolf, H. Prinz, Service Mesh – The New Infrastructure for Microservices, innoQ, 2020, http://leanpub.com/service-mesh-primer More literature is presented each semester according to the task. This also includes online tutorials based on a selection of current frameworks and libraries.
Exams	Laboratory Work 1 Semester (not graded)
Comments	Basic knowledge in the areas of web and component-based distributed systems as well as web and database programming in Java is required. The course includes 50% supervised attendance time (2 SWS) and 50% independent work. Proof of performance is provided by presentation and defense of the solution.

Module Management Competence	
Internal number	INFM140
Coordinator	Prof. Dr. rer. pol. Mathias Philipp
Scope	5.0 ECTS points, 4.0 Contact hours
Placement	1st Semester
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	The module consists of the parts "IT Project Management" and "IT Entrepreneurship". In the IT Project Management course, students are enabled to plan and, if necessary, implement development projects in SWE. To this end, students are familiarized with various industry-specific requirements, methods, process models and tools. The course is held in English. The objective is to prepare students for international IT projects.
Exams	Written Exam 120 Min. (graded)
Lecture IT Project Management	
Internal number	INFM141.a
Lecturer	Prof. Dr. Uwe Haneke
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	English
Content	The lecture will focus on practice oriented project management and highlight special topics like risk or quality management within the context of project management in order to enable students to plan and implement SWE projects - IT-Project Management Process Models - Agile project management - Design Thinking - Requirements specification in IT-projects - Frameworks for scaling agile projects - Risk management - Reporting systems in IT-projects - Process Models for AI and ML projects
Recommended reading	PowerPoint slides, exercise-sheets, case-studies, selected literature on the topic
Exams	Module exam
Comments	Lecture with exercises and case studies
Lecture ITEntrepreneurship	

Internal number	INFM141.b
Lecturer	Prof. Dr. rer. pol. Mathias Philipp
Scope	2.0 ECTS points, 2.0 Contact hours 60 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	Die Vorlesung gliedert sich in die lebenszyklus orientierten Kapitel: - Grundlagen Entrepreneurship - Geschäftsmodell, Geschäftsplanung und unterneherische Handlungsstrategien - Gründungsprozess, Rechtsformen und Rechtsformenvergleich - Businessplan und Finanzierungsarten - Wachstumsmanagement und Unternehmensentwicklung - Unternehmensverkauf Die Studenten sollen unternehmerische Denken und Handeln von der Entwicklung einer Geschäftsidee über die Gründung eines Unternehmens bis zum Verkauf erlernen.
Recommended reading	Lecture material completely on eLearning platform ILIAS and as pdf documents available, blackboard notes for interactive development of central problem positions, LARS-promoted web based training tool for specific preparations and rework of the lectures with multiple choice questions for every process). All together about 160 questions. List of Acronyms and Glossary of Terms, MindMaps for all processes and the complete lecture.
Exams	Module exam
Comments	Teilnahme am seminaristischen Unterricht. Übungen zum Erstellung eines Geschäftsplans, steuerrechtlich korrekten Fakturierung und Auswah der richtigen Gesellschaftsform.

Module Project-based scientific work under supervision 1	
Internal number	INFM150
Coordinator	Prof. Dr. Thomas Fuchß
Scope	5.0 ECTS points, 4.0 Contact hours
Placement	1st Semester
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	On successful completion of the module, students will be able to independently and collaboratively work on scientific topics or application-oriented projects in the IT field. They can apply their academic knowledge to complex challenges, analyze problems, and develop scientifically grounded solutions. Furthermore, they are capable of independently planning and implementing concepts, as well as successfully completing advanced industrial and scientific projects. Through intensive teamwork, students enhance their social and organizational skills and effectively present their results in a collaborative setting.
Exams	Verbal Exam 20 Min. (graded)
Lecture Project work 1	
Internal number	INFM151
Lecturer	Alle Professoren
Scope	5.0 ECTS points, 4.0 Contact hours 150 Stunden gesamt, davon 60 Stunden Kontaktstudium.
Type/mode	Hands-on Experience
Language of instruction	German
Content	Problems and projects may be opened by every faculty member, they are advertized on the message board at the beginning of the term. Scientific research problems usually are at the leading edge of computer science research and may be carried out in cooperation with research institutions. Application projects are of particular relevance for the industrial practice and may be carried out in cooperation with an industrial partner.
Recommended reading	- According to project requirements.
Exams	Hands-on Work 1 Semester (graded)
Comments	- Prerequisites According to project requirements. - Format Presence time and group discussion 30 %, self study 70 %. - Counseling In general, a weekly project session involving the whole team is scheduled.

Module Philosophy of science and ethics	
Internal number	INFM130
Coordinator	Prof. Dr. Thomas Morgenstern
Scope	5.0 ECTS points, 3.0 Contact hours
Placement	2nd Semester
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	After successfully completing this module, students are able to reflect, contextualize and evaluate the theoretical and methodological foundations of computer science and the ethical implications of their professional field. This gives them orientation for their scientific and everyday professional practice. They know different concepts, methods and goals of scientific research, can apply them and can deal with ethical questions of computer science. From these reflective thoughts they can develop their own standards of judgment and for responsible actions as aspiring computer scientists.
Exams	Individual exams
Lecture Philosophy of Science for Computer Science	
Internal number	INFM131
Lecturer	Prof. Dr. Thomas Morgenstern
Scope	3.0 ECTS points, 2.0 Contact hours 90 Stunden gesamt, davon 30 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	Philosophy of science deals with methodologically gained scientific knowledge formulated in theories. It investigates the formation of scientific concepts and structures of theories and the methods used. In particular, we clarify questions such as: "What is knowledge, science, a theory, a description, an explanation, a justification, a confirmation, doubt?", "What are the main currents of the philosophy of science: realism, empiricism, naturalism, critical rationalism, constructivism, philosophy of language, structuralism, holism?", "What is information, a computer, a program?", "How can one methodically acquire knowledge: conceptual, theoretical, logical-deductive, empirical, inductive, simulative, hermeneutic, discursive?", "What is the relationship between scientific theories, reality and the real world?"

Recommended reading	Lecture Script and Notes. Stephan Kornmesser, Wilhelm Büttemeyer: "Wissenschaftstheorie", Springer Verlag. Alan F. Chalmers: "Wege der Wissenschaft", Springer Verlag. Wolfgang Stegmüller: "Probleme und Resultate der Wissenschaftstheorie und Analytischen Philosophie", Springer Verlag. Klaus Mainzer: "Gehirn, Computer, Komplexität", "Computernetze und virtuelle Realität", "The Universe as Automaton", "Künstliche Intelligenz", "Wie berechnebar ist unsere Welt?", "Quantencomputer", "Philosophisches Handbuch Künstliche Intelligenz", "Grenzen der KI", "Zukunft durch nachhaltige Innovation", Springer Verlag. William J. Rapaport: "Philosophy of Computer Science", Wiley. Luciano Floridi: "The Philosophy of Information", Oxford University Press. Depending on a chosen focus, further literature may be recommended.
Exams	Written/verbal Exam/Hands-on Work 60/20/11 Min./Min./Semester (graded)
Comments	The lecture will take the form of a seminar with exercises, presentations and discussions. Different semesters can focus on different areas and address current issues.

Lecture Ethics for computer science

Internal number	INFM132
Lecturer	Prof. Dr. phil. Ziad Mahayni
Scope	2.0 ECTS points, 1.0 Contact hours 60 Stunden gesamt, davon 15 Stunden Kontaktstudium.
Type/mode	Lecture
Language of instruction	German
Content	Information and digital technologies lead to fundamental change in society and the role of humans. This is accompanied by new and far-reaching ethical questions such as the handling of artificial intelligence, the future of human work in the context of advancing automation, the monitoring and manipulation of human behaviour, the creation of new habitats in virtual worlds, and the technical optimization of humans (trans/posthumanism). The course explores these ethical questions, discusses response options and evaluates ethical issues. Finally, the influence of computer science on these developments is reflected and ethical guidelines for the development of information technologies are discussed.

Recommended reading	Lecture Script and Notes. Julian Nida-Rümelin, Nathalie Weidenfeld: Digitaler Humanismus. Eine Ethik für das Zeitalter der Künstlichen Intelligenz, Piper Verlag Sarah Spiekermann: Digitale Ethik: Ein Wertesystem für das 21. Jahrhundert, Droemer Verlag Cathy O'Neil: Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy, Penguin Random House Michael Hauskeller, Mythologies of Transhumanism, Palgrave Macmillan Guidelines for the ethics of Artificial Intelligence such as those formulated by the European Union, UNESCO or even by companies. Depending on a chosen focus, further literature may be recommended.
Exams	Written/verbal Exam/Hands-on Work 60/20/1 Min./Min./Semester (graded)
Comments	The lecture will take the form of a seminar with exercises, presentations and discussions. Different semesters can focus on different areas of focus and address current issues.

Module Project-based scientific work under supervision 2	
Internal number	INFM240
Coordinator	Prof. Dr. Thomas Fuchß
Scope	5.0 ECTS points, 4.0 Contact hours
Placement	2nd Semester
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	On successful completion of the module, students will be able to apply advanced techniques and methods of scientific and project-based work. They can independently and collaboratively apply their acquired knowledge to more complex and large-scale challenges. Students have developed the ability to critically evaluate decisions, consider alternative approaches, and assess and justify their own choices. They are capable of independently planning, executing, and successfully completing advanced and large-scale industrial and scientific IT projects. Within a team, they take on roles, work effectively together, and take responsibility for the success of the project.
Exams	Verbal Exam 20 Min. (graded)
Lecture Project work 2	
Internal number	INFM241
Lecturer	Alle Professoren
Scope	5.0 ECTS points, 4.0 Contact hours 150 Stunden gesamt, davon 60 Stunden Kontaktstudium.
Type/mode	Hands-on Experience
Language of instruction	German
Content	Problems and projects may be opened by every faculty member, they are advertized on the message board at the beginning of the term. Scientific research problems usually are at the leading edge of computer science research and may be carried out in cooperation with research institutions. Application projects are of particular relevance for the industrial practice and may be carried out in cooperation with an industrial partner.
Recommended reading	- According to project requirements.
Exams	Hands-on Work 1 Semester (graded)
Comments	- Prerequisites According to project requirements. - Format Presence time and group discussion 30 %, self study 70 %. - Counseling In general, a weekly project session involving the whole team is scheduled.

Module Advanced seminar	
Internal number	INFM250
Coordinator	Prof. Dr. rer. pol. Mathias Philipp
Scope	5.0 ECTS points, 4.0 Contact hours
Placement	2nd Semester
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	
Exams	Verbal Exam 20 Min. (graded)
Lecture Seminar with presentation	
Internal number	INFM251
Lecturer	Alle Professoren
Scope	5.0 ECTS points, 4.0 Contact hours 150 Stunden gesamt, davon 60 Stunden Kontaktstudium.
Type/mode	Thesis
Language of instruction	German
Content	
Recommended reading	
Exams	Presentation 1 Semester (graded)
Comments	

Module Thesis with Colloquium	
Internal number	INFM310
Coordinator	Prof. Dr. Heiko Körner
Scope	30.0 ECTS points, 0.0 Contact hours
Placement	3rd Semester
Pre-requisites with regard to content	none
Pre-requisites according to the examination regulations	none
Competences	The Master's thesis is the final task of the Computer Science degree programme. Participation in this module enables students to work on a scientific or application-related topic largely independently. In co-operation with an industrial company or research institution, they are able to carefully structure the task, collect the necessary resources and then solve the task on the basis of a specially derived schedule. They can both document their results in writing and present and defend them in a colloquium in front of a specialist audience in a scientifically appealing manner.
Exams	Individual exams
Lecture Thesis	
Internal number	INFM311
Lecturer	Alle Professoren
Scope	29.0 ECTS points, 0.0 Contact hours 870 Stunden gesamt, davon 0 Stunden Kontaktstudium.
Type/mode	Thesis
Language of instruction	German
Content	In the final thesis, students work independently on a practical problem or research task within a specified period of time using scientific methods. They structure the task, check dependencies, collect the necessary resources and work on the problem according to a timetable. The written thesis summarises the results in a didactically meaningful way and meets academic standards.
Recommended reading	Suitable for the task as agreed
Exams	Master Thesis 6 Months (graded)
Comments	
Lecture Thesis Defense	
Internal number	INFM312
Lecturer	Alle Professoren
Scope	1.0 ECTS points, 0.0 Contact hours 30 Stunden gesamt, davon 0 Stunden Kontaktstudium.
Type/mode	Colloquium
Language of instruction	German

Content	The final examination covers all topics relevant to computer science in the Master's degree programme. Students demonstrate that they have understood and can apply interdisciplinary contexts. They present the most important results of their Master's thesis to a specialised audience in a didactically appropriate manner. In addition, they answer questions from various areas of computer science that are related to their thesis. With the final examination, they prove that they have the competence to independently work on novel problems in computer science.
Recommended reading	
Exams	Verbal Exam 30 Min. (not graded)
Comments	