Trier University
of Applied Sciences

HOCHSCHULE TRIER

MODULHANDBUCH

der Master-Studiengänge im Fachbereich Informatik
Prüfungsordnungen 2027

MODULE HANDBOOK

for the Master's programmes
in the Department of Computer Science
Examination Regulations 2027

Revision: 29.07.2026

Hinweise zu den Modulhandbucheinträgen

- Die Dauer aller Module beträgt jeweils ein Semester.
- Die zeitliche Lage aller Module ergibt sich aus den Anlagen der Fachprüfungsordnungen.
- Neben den jeweils genannten Voraussetzungen werden in allen Modulen Grundkenntnisse der Informatik erwartet, die in der Regel in einem Bachelor-Studium der Informatik vermittelt werden.
- Zur individuellen Vorbereitung kann der Besuch der im Feld „Empfohlene Voraussetzungen" genannten Module dienen, ebenso wie die Angaben im Feld „Literatur". Weitere Literaturangaben finden sich ggf. in den Vorlesungsunterlagen, die i.d.R. auf Stud.IP (https://studip.hochschule-trier.de) zur Verfügung gestellt werden.
- Falls in einem Modul eine Studienleistung zu erbringen ist, ist diese gemäß § 6 der Allgemeinen Prüfungsordnung Voraussetzung für die Zulassung zur Prüfung. Die semesteraktuelle Form der Studienleistung wird zu Beginn der Lehrveranstaltung bekannt gegeben.
- Bei Angabe mehrerer alternativer Prüfungsformen für ein Modul wird die semesteraktuelle Prüfungsform zu Beginn der Lehrveranstaltung bekannt geben.
- Voraussetzung für die Vergabe von ECTS-Punkten ist eine Bewertung der Prüfungsleistung mit mindestens der Note „ausreichend".
- Die Gewichtung eines Modulergebnisses zur Bildung der Gesamtnote ergibt sich gemäß § 10 der Fachprüfungsordnungen aus dem Verhältnis der ECTS-Punkte für das Modul und der Gesamtanzahl der ECTS-Punkte.
- Dieses Modulhandbuch spiegelt das jeweils aktuelle Lehrangebot wider. Die Modulhandbucheinträge nicht mehr angebotener Module können im Fachbereich Informatik angefragt werden.
- Neben den oben genannten Modulen könnten weitere Module von anderen Fachbereichen angeboten werden, welche in den Master-Studiengängen des Fachbereichs Informatik als Pflicht- oder Wahlpflichtmodul zur Verfügung stehen.

Notes on the module handbook entries

- All modules last one semester each.
- The timetable for all modules is set out in the appendices to the examination regulations.
- In addition to the specific prerequisites listed for each module, all modules require a basic knowledge of computer science, which is generally acquired during a Bachelor's degree in Computer Science.
- To help with individual preparation, students may wish to attend the modules listed in the "Recommended prerequisites" field, as well as consult the information in the "Reading list" field. Further reading recommendations may be found in the lecture material, which is usually made available on Stud.IP (https://studip.hochschule-trier.de).
- If a module requires the completion of an examination prerequisite, this is a prerequisite for admission to the examination in accordance with section 6 of the general examination regulations. The form of the examination prerequisite applicable for the current semester will be announced at the start of the course.
- Where several alternative examination formats are specified for a module, the examination format applicable for the current semester will be announced at the start of the course.
- A minimum grade of "satisfactory" must be achieved in the examination in order for ECTS credits to be awarded.
- The weighting of a module result in calculating the overall classification is determined, in accordance with section 10 of the examination regulations, by the ratio of the ECTS credits for the module to the total number of ECTS credits.
- This module handbook reflects the current range of modules on offer. Module handbook entries of modules that are no longer offered can be obtained from the Department of Computer Science.
- In addition to the modules listed above, further modules may be offered by other departments, which are available as compulsory or elective modules within the Master's programmes of the Department of Computer Science.

Inhaltsverzeichnis · Table of Contents

Abschlussarbeit mit Kolloquium (D) 6
Academic Communication in German (E) 7
Advanced Programming (E) 8
Advanced Symbolic AI (E) 10
AI-aided Software Engineering (D) 11
AI-aided Software Engineering (E) 12
Anforderungsmanagement (D) 13
Artificial Intelligence for Robots (E) 15
Berechenbarkeit und Komplexität (D) 16
Cognitive Systems (E) 18
Cyber Security (E) 19
Data Science (D) 21
Data Science (E) 23
Data Warehouse (D) 25
Ethical and Responsible AI (D) 26
Ethical and Responsible AI (E) 28
Fortgeschrittene Mensch-Computer-Interaktion (D) 30
Fortgeschrittene Programmierkonzepte (D) 32
Fortgeschrittene Symbolische KI (D) 34
Generative AI (E) 35
Generative KI (D) 36
German Language and Culture (E) 37
High Performance Computing (E) 38
Informationssicherheit (D) 39
Interactive Physical Simulation (E) 41
Knowledge Representation with Description Logics (E) 42
Learning Analytics and Educational Data Mining (D) 43
Machine Learning and Neural Networks (E) 44
Maschinelles Lernen und Neuronale Netze (D) 45
Master's Thesis with Oral Defence (E) 46
Mixed Reality (D) 47
Mobile Robotics (E) 49
Optimierung (D) 50
Professional Communication in German (E) 52
Project (E) 54
Projektarbeit (D) 55
Safety KI-basierter Cyber Physical Systems (D) 56

Seminar (D) **57**
Seminar (E) **58**
Simulation Technology and Reinforcement Learning (E) **59**
Simulationstechnik und Reinforcement Learning (D) **60**
Software-Architekturen (D) **61**
Software-Qualitätsmanagement (D) **62**
Statistics and Learning Theory (D) **63**
Statistics and Learning Theory (E) **64**
Verlässliche Echtzeitsysteme (D) **65**
Visual Navigation (E) **66**

Abschlussarbeit mit Kolloquium (D)				
Inhalte	Die Abschlussarbeit umfasst die Bearbeitung einer qualifizierten Aufgabenstellung deren Schwierigkeitsgrad der späteren Berufspraxis eines Master-Absolventen entspricht mithilfe wissenschaftlicher Methoden. Die Aufgabenstellung kann theoretische, experimentelle, empirische oder praxisorientierte Probleme umfassen. Dabei werden systematische Vorgehensweisen und sinnvolle Arbeitstechniken eingeübt sowie die Verbindung zu Anwendungsgebieten der Informatik hergestellt. Die Studierenden präsentieren ihre Ergebnisse in einer Ausarbeitung unter Anwendung der Kriterien wissenschaftlichen Arbeitens sowie in einem Kolloquium vor einer Prüfungskommission. Dabei wird der Inhalt der Abschlussarbeit im Kontext des jeweiligen Themas hinterfragt.			
Lernergebnisse	Die Studierenden haben durch die erfolgreiche Bearbeitung gezeigt, dass sie in der Lage sind, innerhalb einer vorgegebenen Frist ein größeres Fachproblem selbstständig mit wissenschaftlichen Methoden zu bearbeiten. Sie verfügen über ein breites und integriertes Wissen, einschließlich der wissenschaftlichen Grundlagen sowie über ein kritisches Verständnis der wichtigsten Theorien und Methoden. Sie sind in der Lage, die im Studium erworbenen Kenntnisse, Fähigkeiten, Methoden und Werkzeuge auf neue Fragestellungen zu übertragen und darüber hinaus selbstständig, um relevante Inhalte zu erweitern, zu bewerten und wissenschaftlich zu interpretieren. Sie leiten auf dieser Basis fundierte Lösungsansätze ab und formulieren eine dem Stand der Wissenschaft entsprechende Lösung für das Fachproblem. Sie können ihre Ergebnisse darüber hinaus professionell und unter Anwendung der Kriterien des wissenschaftlichen Arbeitens dokumentieren, in einem Kolloquium darlegen und argumentativ vertreten.			
Lehrform	☐ Vorlesung			
	☐ Übung			
	☐ Seminar/Seminaristischer Unterricht			
	☐ Labor			
	☒ Projekt			
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Module „Seminar" und „Projektarbeit". Weitere empfohlene Voraussetzungen sind abhängig von der Aufgabenstellung und werden vom Betreuer festgelegt.			
Literatur	▪ Helmut Balzert, Marion Schröder, Christian Schäfer: Wissenschaftliches Arbeiten. W3L GmbH. ▪ Booth, Wayne C.; Colomb, Gregory G.; Williams, Joseph M.; Bizup, Joseph; Fitzgerald, William T. (2016): The Craft of Research. 4. Auflage. University of Chicago Press. ▪ Zobel, Justin (2014): Writing for Computer Science. 3. Auflage. Springer. ▪ Turabian, Kate L.; Booth, Wayne C.; Colomb, Gregory G.; Williams, Joseph M.; Bizup, Joseph (2018): A Manual for Writers of Research Papers, Theses, and Dissertations. 9. Auflage. University of Chicago Press. ▪ Evans, David; Gruba, Paul; Zobel, Justin (2014): How to Write a Better Thesis. 3. Auflage. Springer.			
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)			
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)			
	☐ Bestehen von Leistungsstandkontrollen			
Prüfungsform	☐ Mündliche Prüfung			
	☐ Klausur			
	☐ Prüfung am PC			
	☐ Hausarbeit (ggf. mit Präsentation)			
	☒ Projekt (ggf. mit Präsentation)			
Verwendbarkeit	Informatik		☒ Pflicht ☐ Wahlpflicht	
	Künstliche Intelligenz und Data Science		☒ Pflicht ☐ Wahlpflicht	
	Artificial Intelligence in Computer Science		☒ Pflicht ☐ Wahlpflicht	
Angebot	☒ Sommersemester ☒ Wintersemester ☐ Unregelmäßig			
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium	
	25	30 Stunden	720 Stunden	
Lehrende(r)	Dozenten des Fachbereichs Informatik			
Modulverantwortliche(r)	Fachrichtungsleiter Informatik			
Änderungsdatum	18.06.2026			

Academic Communication in German (E)				
Contents	▪ Development of German language skills for academic contexts, building on A2 and leading towards B1. ▪ Academic vocabulary and structures: describing study programmes, courses, projects, data, processes, diagrams and simple technical topics in computer science. ▪ Grammar and text competence: subordinate clauses, conjunctions, perfect and past tense use, adjective endings, passive constructions, nominalisation, connectors and paragraph structure. ▪ Reading strategies for academic and university-related texts: course descriptions, examination regulations, abstracts, short technical articles, instructions and administrative documents. ▪ Writing practice: formal and semi-formal e-mails, summaries, short explanations, project descriptions, learning journals and simple argumentation. ▪ Speaking and listening: participating in discussions, asking questions in class, giving short presentations, explaining a project idea and responding to feedback.			
Learning outcomes	Upon completion of the course, students will be able to ▪ communicate in German in recurring academic situations such as courses, consultations, group work and university administration. ▪ understand the main points of clear standard German in academic and student-related contexts when topics are familiar. ▪ read and summarise short academic or technical texts and extract relevant information for study-related tasks. ▪ write structured e-mails, short summaries and brief academic texts using appropriate register and basic subject-specific vocabulary. ▪ give a short, comprehensible presentation on a study-related or technical topic and answer simple follow-up questions. ▪ use strategies for autonomous language learning, vocabulary acquisition and preparation for B1-level communication.			
Learning format	☐ Lecture			
	☒ Exercise			
	☒ Seminar/Seminar-style class			
	☐ Lab			
	☒ Project			
Recommended prerequisites	Competencies in accordance with the learning outcomes of the module "German Language and Culture"			
Reading list	▪ Aspekte neu B1 plus or equivalent CEFR-oriented course material for academic communication ▪ Mit Erfolg zu telc Deutsch B1 / Goethe-Zertifikat B1 or comparable practice material ▪ University-specific materials: module descriptions, examination regulations, project briefs and administrative templates ▪ Selected short texts from computer science and applied AI contexts adapted to B1 level			
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)			
	☐ Completion of term paper/lab assignments (with a presentation, if applicable)			
	☒ Passing proficiency assessments			
Examination format	☒ Oral exam			
	☐ Written exam			
	☐ Computer-based exam			
	☐ Term paper (with a presentation, if applicable)			
	☐ Project (with a presentation, if applicable)			
Eligibility	Computer Science	☐ Compulsory ☐ Elective		
	Artificial Intelligence and Data Science	☐ Compulsory ☐ Elective		
	Artificial Intelligence in Computer Science	☒ Compulsory ☐ Elective (non-native speakers only)		
Offer	☒ Summer term ☐ Winter term ☐ Irregular			
Workload	ECTS credits	Contact time	Self-study	
	5	60 hours	90 hours	
Lecturer	Contract lecturer			
Person in charge	Prof. Dr. Georg Schneider			
Date of change	08.07.2026			

Advanced Programming (E)		
Contents	This course covers advanced programming concepts and their application in modern programming languages. Building on prior knowledge of programming and software development, the module explores concepts such as ▪ abstraction ▪ advanced type systems ▪ functional programming ▪ concurrency models ▪ metaprogramming ▪ memory management ▪ language runtime systems ▪ advanced data structures and ▪ algorithmic design techniques Students develop a deep understanding of the principles underlying modern programming languages and learn to systematically analyse and compare programming paradigms, language features, design approaches, and algorithmic solutions. The course combines theoretical foundations with practical programming examples and exercises, including the implementation and evaluation of algorithms and data structures on large datasets. Python serves as the primary implementation language, while concepts are discussed in a language-independent manner and related to features found in a wide range of modern programming languages.	
Learning outcomes	Upon completion of the course, students will be able to ▪ explain and apply advanced programming concepts and language features in practical programming tasks, ▪ analyse and compare programming paradigms, language designs, implementation approaches, and algorithmic solutions, ▪ design, implement, and evaluate advanced data structures and algorithms for non-trivial programming problems, ▪ reason about program behavior, correctness, performance, scalability, concurrency, and memory management using appropriate conceptual models, ▪ evaluate trade-offs between alternative programming techniques, data structures, algorithms, and language features, ▪ transfer advanced programming concepts to unfamiliar programming languages, frameworks, and technologies, ▪ communicate and discuss programming-related concepts, algorithmic design decisions, and implementation trade-offs using precise technical terminology.	
Learning format	☒ Lecture	
	☒ Exercise	
	☐ Seminar/Seminar-style class	
	☒ Lab	
	☐ Project	
Recommended prerequisites	Competencies in accordance with the learning outcomes of the Bachelor's modules "Introduction to Programming", "Foundations of Object-oriented Programming", "Advanced Object-oriented Programming", "Software Design" and "Data Structures and Algorithms", furthermore basic knowledge of complexity theory such as runtime analysis and basic complexity classes.	
Reading list	▪ Ramalho, L.: Fluent Python: Clear, Concise, and Effective Programming (2nd ed.). O'Reilly Media, 2022. ▪ Viafore, P.: Robust Python: Write Clean and Maintainable Code (1st ed.). O'Reilly Media, 2021. ▪ Van Roy, P., & Haridi, S.: Concepts, Techniques, and Models of Computer Programming (1st ed.). MIT Press, 2004. ▪ Scott, M. L., & Aldrich, J.: Programming Language Pragmatics (5th ed.). Morgan Kaufmann / Elsevier, 2025. ▪ Cormen et al: Introduction to Algorithms (3rd Edition), 2009, The MIT Press ▪ Hetland: Python Algorithms, 2010, APress	
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)	
	☐ Completion of term paper/lab assignments (with a presentation, if applicable)	
	☒ Passing proficiency assessments	
Examination format	☐ Oral exam	
	☒ Written exam	
	☐ Computer-based exam	
	☐ Term paper (with a presentation, if applicable)	
	☐ Project (with a presentation, if applicable)	
Eligibility	Computer Science	☒ Compulsory ☐ Elective

	Artificial Intelligence and Data Science		☒ Compulsory ☐ Elective
	Artificial Intelligence in Computer Science		☒ Compulsory ☐ Elective
Offer	☐ Summer term ☒ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Prof. Dr. Thomas Hartmann, Prof. Dr. Heinz Schmitz		
Person in charge	Prof. Dr. Thomas Hartmann		
Date of change	07.07.2026		

Advanced Symbolic AI (E)			

Contents	▪ Knowledge representation and automated reasoning
	  ▪ Formal modelling of knowledge, rules and relationships
	  ▪ Derivation of implicit knowledge from explicit knowledge bases
	▪ SAT- and SMT-based modelling and problem-solving
	  ▪ Solver-based approaches to decision-making, planning and configuration problems
	▪ Non-monotonic reasoning
	  ▪ Reasoning with incomplete, changing or contradictory knowledge
	  ▪ Answer Set Programming
	▪ Knowledge graphs as a structured knowledge base for symbolic reasoning
	▪ Explanations and justifications for symbolic inferences
	▪ Combination of symbolic and data-driven methods, in particular neuro-symbolic reasoning		

Learning outcomes	Upon completion of the course, students will be able to ▪ to explain approaches to knowledge representation and to formally model knowledge, rules and relationships. ▪ to apply automated reasoning methods to derive implicit knowledge from explicit knowledge bases. ▪ to classify SAT- and SMT-based approaches for modelling and solving decision-making, planning and configuration problems, and to apply them to selected problems. ▪ to explain non-monotonic inference methods and to use them for dealing with incomplete, changing or contradictory knowledge. ▪ describe knowledge graphs as structured knowledge bases for symbolic reasoning and use basic query and inference capabilities. ▪ understand explanations and justifications for symbolic inferences and reflect on their significance for hybrid and neuro-symbolic AI systems.		
Learning format	☒ Lecture		
	☒ Exercise		
	☐ Seminar/Seminar-style class		
	☐ Lab		
	☐ Project		
Recommended prerequisites	None		
Reading list	▪ Stuart Russell, Peter Norvig. Artificial Intelligence: A Modern Approach. Fourth Edition, Pearson, 2020 ▪ Barrett, C. et al. (2025). Satisfiability Modulo Theories: A Beginner's Tutorial. In: Platzer, A., Rozier, K.Y., Pradella, M., Rossi, M. (eds) Formal Methods. FM 2024. Lecture Notes in Computer Science, vol 14934. Springer, Cham ▪ Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Torsten Schaub: Answer Set Solving in Practice. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers 2012		
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)		
	☐ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☒ Passing proficiency assessments		
Examination format	☐ Oral exam		
	☒ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☐ Project (with a presentation, if applicable)		
Eligibility	Computer Science	☐ Compulsory ☒ Elective	
	Artificial Intelligence and Data Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence in Computer Science	☒ Compulsory ☐ Elective	
Offer	☒ Summer term ☐ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Prof. Dr. Claudia Schon		
Person in charge	Prof. Dr. Claudia Schon		
Date of change	29.06.2026		

AI-aided Software Engineering (D)			
Inhalte	Die Lehrveranstaltung bietet eine Einführung in KI-gestützte Werkzeuge und Prozesse für alle Phasen des Softwareentwicklungszyklus, darunter (unter anderem) Anforderungsanalyse, Softwareentwurf und -architektur, Programmierung, Qualitätssicherung und Dokumentation. Auf der Grundlage eines umfassenden Überblicks über relevante Konzepte und Technologien setzen sich die Studierenden mit den Vorteilen, Chancen und möglichen Nachteilen der KI-gestützten Softwareentwicklung auseinander. Neben theoretischen Überlegungen und Diskussionen im Unterricht wenden die Studierenden KI-gestützte Werkzeuge und Prozesse systematisch in kleinen Projekten an, um praktische Erfahrungen in realistischen Szenarien zu sammeln.		
Lernergebnisse	Die Studierenden sind nach der Teilnahme in der Lage ▪ KI-gestützte Programmierwerkzeuge (Code-Generierung, Refactoring, Debugging) effektiv und kritisch einzusetzen – und dabei zu beurteilen, wann sie den Ergebnissen der KI vertrauen können und wann sie diese hinterfragen sollten; ▪ die Rolle der KI bei Entscheidungen zur Softwarearchitektur und zum Softwareentwurf zu bewerten und zu erläutern, was KI beim Systemdesign leisten kann und was nicht; ▪ einen Entwicklungs-Workflow (agiler Prozess, CI/CD, Dokumentation) neu zu gestalten, um KI-gestützte Werkzeuge angemessen zu integrieren; ▪ Qualität und Risiken von KI-generiertem Code zu bewerten – einschließlich Bedenken hinsichtlich Sicherheit, Tests, Lizenzierung und übermäßiger Abhängigkeit; ▪ kritisch zu reflektieren, wie KI die Rolle, die Verantwortlichkeiten und die erforderlichen Kompetenzen von Softwareentwicklerinnen und Softwareentwicklern verändert; ▪ KI-gestützte Ansätze mit traditionellen Ansätzen anhand konkreter Beispiele aus der eigenen Projektarbeit zu vergleichen.		
Lehrform	☒ Vorlesung		
	☐ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☒ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	Wird zu Beginn der Lehrveranstaltung bekanntgegeben		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik	☒ Pflicht ☐ Wahlpflicht	
	Künstliche Intelligenz und Data Science	☒ Pflicht ☐ Wahlpflicht	
	Artificial Intelligence in Computer Science	☒ Pflicht ☐ Wahlpflicht	
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Thomas Hartmann, Prof. Dr. Michael Striewe		
Modulverantwortliche(r)	Prof. Dr. Michael Striewe		
Änderungsdatum	08.07.2026		

AI-aided Software Engineering (E)			
Contents	The course introduces AI-aided tools and processes for all stages of the software engineering lifecycle, including (but not limited to) requirements engineering, software design and architecture, coding, quality assurance and documentation. Based on a comprehensive overview on relevant concepts and technologies, students will reflect upon benefits, chances and potential drawbacks of AI-aided Software Engineering. Besides theoretical considerations and in-class discussions, students will apply AI-aided tools and processes systematically in small projects to gather hands-on experience in realistic settings.		
Learning outcomes	Upon completion of the course, students will be able to ▪ apply AI coding tools (code generation, refactoring, debugging) effectively and critically — knowing when to trust and when to question AI output; ▪ evaluate AI's role in software architecture and design decisions and articulate what AI can and cannot do in system design; ▪ redesign a development workflow (agile process, CI/CD, documentation) to integrate AI tools appropriately; ▪ assess quality and risk in AI-generated code — including security, testing, licensing, and over-reliance concerns; ▪ reflect critically on how AI shifts the engineer's role, responsibilities, and required skills; ▪ compare AI-assisted vs. traditional approaches with concrete evidence from their own project work.		
Learning format	☒ Lecture		
	☐ Exercise		
	☐ Seminar/Seminar-style class		
	☐ Lab		
	☒ Project		
Recommended prerequisites	None		
Reading list	To be announced at the start of the course		
Examination prerequisite	☐ Regular completion of homework assignments (with a presentation, if applicable)		
	☒ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☐ Passing proficiency assessments		
Examination format	☐ Oral exam		
	☒ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☐ Project (with a presentation, if applicable)		
Eligibility	Computer Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence and Data Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence in Computer Science	☒ Compulsory ☐ Elective	
Offer	☒ Summer term ☐ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Prof. Dr. Thomas Hartmann, Prof. Dr. Michael Striewe		
Person in charge	Prof. Dr. Michael Striewe		
Date of change	08.07.2026		

Anforderungsmanagement (D)			
Inhalte	In den meisten Unternehmen sind Anforderungen an Softwaresysteme oft unklar, widersprüchlich, unvollständig und nicht nachvollziehbar dokumentiert. Existierende Anforderungsspezifikationen (Lasten- und Pflichtenhefte) sind veraltet. Wichtige Anforderungen werden oft zu spät erkannt oder sogar übersehen. Darüber hinaus werden Anforderungen oft qualitativ unzureichend formuliert und lassen Spielraum für Interpretation. Die Folgen: unzufriedene Kunden, explodierende Kosten, weit überschrittene Projekttermine, unwartbare Systeme. Aufgabe des Anforderungsmanagements (engl.: Requirements Engineering oder RE) ist es, aus oft vagen und teilweise widersprüchlichen Ideen eine möglichst vollständige, korrekte widerspruchs- und redundanzfrei, nachverfolgbare und atomare Systemspezifikationen zu erzeugen, um diesen aufgeführten Problemen frühzeitig entgegenwirken zu können. Im zweiten Teil der Vorlesung wird auf aktuelle Entwicklungen im Bereich des Anforderungsmanagement eingegangen. Hierbei wird der Zusammenhang zwischen Anforderungsmanagement und Variantenmanagement behandelt. Im Mittelpunkt steht zunächst das Verstehen der variantenbezogenen Entwicklungstechniken (Domänen- und Applikationsengineering). Das Wissen, Verstehen und die Anwendung werden durch praxisorientierte Übungen, sowie den Einsatz aktueller Anforderungsmanagement- und Variantenmanagement-Werkzeuge vertieft.		
Lernergebnisse	Die Studierenden haben aufbauend auf der gleichnamigen Bachelor-Vorlesung die Kenntnisse bzgl. der Modellierung, Ermittlung und Dokumentation von Anforderungen vertieft. Die Studierenden können ▪ Systeme hinsichtlich Ihrer wesentlichen Eigenschaften erfassen und abstrahieren, ▪ Anforderungen (funktional, nicht-funktional) an Systeme sammeln und spezifizieren, ▪ Techniken zur Stakeholder-Identifikation und Interviews kennen und anwenden, ▪ Review-Arten unterscheiden und anwenden, ▪ Anforderungen auf deren Qualität beurteilen, ▪ Agile Vorgehensweisen im Requirements Engineering unterscheiden und anwenden, ▪ aktuelle Entwicklungen wie Standardisierungen im Anforderungsmanagement (z.B. ReqIF) beurteilen, und ▪ aktuelle Werkzeuge aus dem Bereich Variantenmanagement im Requirements Engineering anwenden.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☒ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „Grundlagen des Anforderungsmanagements“		
Literatur	▪ Ian Sommerville: Software Engineering. Pearson Studium, 2007. ▪ Klaus Pohl: Requirements Engineering - Grundlagen, Prinzipien, Techniken. dpunkt.Verlag, 2. Auflage, 2008. ▪ Soren Lauesen: Software Requirements, Styles and Techniques. Addison-Wesley, 2002. ▪ Colin Hood, Simon Wiedemann, Stefan Fichtinger, Urte Pautz: Requirements Management. Springer, 2008. ▪ Helmuth Partsch: Requirements-Engineering systematisch. Springer, 2. Auflage, 2010.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik	☒ Pflicht ☐ Wahlpflicht	
	Künstliche Intelligenz und Data Science	☐ Pflicht ☒ Wahlpflicht	
	Artificial Intelligence in Computer Science	☐ Pflicht ☒ Wahlpflicht	
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Georg Rock		

Modulverantwortliche(r)	Prof. Dr. Georg Rock
Änderungsdatum	17.06.2026

Artificial Intelligence for Robots (E)			
Contents	▪ Decision-making and planning for intelligent robots and cognitive systems ▪ Probabilistic reasoning ▪ Inference ▪ Parameter estimation ▪ Bayesian networks and structure learning ▪ Simple and complex decisions ▪ Sequential problems ▪ Decision-making under uncertainty ▪ Model-based and model-free methods ▪ Behaviour under state uncertainty ▪ Multi-agent systems and collaboration		
Learning outcomes	Students will be able to ▪ develop and implement stochastic models, ▪ understand what Bayesian networks are and apply them to various problems, ▪ apply Bayesian parameter estimation and derive exploration strategies, ▪ understand and implement MAP estimators, ▪ apply MDPs and POMDPs, ▪ understand and apply Q-learning, ▪ formulate dynamic programs, ▪ classify search strategies and use them efficiently, ▪ apply local optimisation strategies and supplement them with heuristics, and ▪ where appropriate, understand planning methods in theory and apply them in practice.		
Learning format	☒ Lecture		
	☒ Exercise		
	☐ Seminar/Seminar-style class		
	☐ Lab		
	☐ Project		
Recommended prerequisites	A basic understanding of stochastics and machine learning is an advantage, but is not a prerequisite.		
Reading list	▪ S. M. LaValle: Planning Algorithms. Cambridge University Press, 2006. ▪ M. J. Kochenderfer et al.: Algorithms for Decision Making. MIT Press, 2022. ▪ R. Neapolitan: Learning Bayesian Networks. Prentice Hall, 2003. ▪ S. Russell and P. Norvig: Artificial Intelligence: A Modern Approach. Pearson, 4. Auflage (2020).		
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)		
	☒ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☒ Passing proficiency assessments		
Examination format	☒ Oral exam (if there are few participants)		
	☒ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☐ Project (with a presentation, if applicable)		
Eligibility	Computer Science		☐ Compulsory ☒ Elective
	Artificial Intelligence and Data Science		☐ Compulsory ☒ Elective
	Artificial Intelligence in Computer Science		☐ Compulsory ☒ Elective
Offer	☒ Summer term ☐ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Prof. Dr. Jürgen Graf		
Person in charge	Prof. Dr. Jürgen Graf		
Date of change	06.07.2026		

Berechenbarkeit und Komplexität (D)			
Inhalte	▪ Äquivalente Modelle der Computer-Berechenbarkeit: WHILE, Turingmaschinen ▪ Algorithmenbegriff, These von Church ▪ Berechenbare Funktionen und ihre grundlegenden Eigenschaften, z.B. s-m-n-Theorem, universelle Algorithmen, Gödelisierung ▪ Entscheidbarkeit, Semi-Entscheidbarkeit Aufzählbarkeit, äquivalente Charakterisierungen, die Klassen REC und RE ▪ Unentscheidbare Mengen, Halteproblem, Many-one-Reduzierbarkeit, RE-Vollständigkeit, Beispiel nicht-aufzählbarer Mengen, Reduktionsmethode, Satz von Rice ▪ Laufzeitanalyse, Polynomial- und Exponentialzeit, Katalog klassischer Entscheidungsprobleme ▪ Die Klassen P und NP, NP-Vollständigkeit, originärer Vollständigkeitsbeweis ▪ Reduktionen durch Komponentendesign, Vollständigkeitsbeweise für Auswahl-, Graphen- und Reihenfolgeprobleme ▪ Strenge NP-Vollständigkeit, Pseudo-Polynomialzeit, Large Number Problems, Dynamische Programmierung, pseudo-polynomielle Reduktionen ▪ NP-Optimierungsprobleme, die Klassen PO und NPO, Turing-Reduzierbarkeit, Begriff der (strengen) NP-Härte, PO-NPO-Frage ▪ Approximationsalgorithmen, Performanzrate, Beispiele mit konstanter Performanzrate, Approximierbarkeit, Approximationsschema		
Lernergebnisse	Die Studierenden können ▪ Berechenbarkeitsmodelle hinsichtlich ihrer prinzipiellen Leistungsfähigkeit einordnen, ▪ die Bedeutung eines präzisen Algorithmenbegriffs erläutern, ▪ Berechnungsprobleme hinsichtlich ihrer prinzipiellen algorithmischen Beherrschbarkeit analysieren und bewerten, ▪ die Bedeutung der P-NP-Theorie präsentieren und Berechnungsprobleme in dieser Hinsicht klassifizieren, ▪ die Methode der Reduktion auf neue Berechnungsprobleme anwenden und ▪ Optimierungsprobleme kategorisieren und deren praktische Lösbarkeit beurteilen.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ H. Rogers Jr.: Theory of Recursive Functions and Effective Computability. MIT–Press, 1967. 3rd printing 1992. ▪ Michael R. Garey and David S. Johnson: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., 1979. ▪ K. W. Wagner: Theoretische Informatik – Eine kompakte Einführung. Springer, 2nd edition, 2003.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik	☒ Pflicht ☐ Wahlpflicht	
	Künstliche Intelligenz und Data Science	☐ Pflicht ☒ Wahlpflicht	
	Artificial Intelligence in Computer Science	☐ Pflicht ☒ Wahlpflicht	
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Heinz Schmitz		
Modulverantwortliche(r)	Prof. Dr. Heinz Schmitz		

Änderungsdatum	17.06.2026

Cognitive Systems (E)			
Contents	▪ Introduction to cognitive systems and sensors (IMU, GNSS, camera, LiDAR) ▪ LiDAR odometry (ICP, GICP, NDT) ▪ Image formation and epipolar geometry (E-matrix, F-matrix) ▪ Stereo visual odometry ▪ Deep learning (backpropagation, CNNs) ▪ Attention mechanisms and vision transformers ▪ (Graph-based) knowledge representation ▪ Decision making ▪ Robot kinematics and URDF ▪ INS/GNSS and strapdown algorithm ▪ Kalman filtering (KF, EKF) ▪ Control (PID, LQR)		
Learning outcomes	Upon completion of the course, students will be able to ▪ understand and apply multi-modal sensor models and calibration methods; ▪ derive and implement geometric vision algorithms including stereo depth estimation, feature tracking, and epipolar geometry; ▪ design, train, and evaluate deep neural networks for perception tasks; ▪ represent, query, and update structured world knowledge using graph models and Bayesian inference; ▪ apply optimal decision-making methods in single-agent and multi-agent settings; ▪ model robot kinematics and implement closed-loop control strategies; ▪ implement and tune Kalman filters for multi-sensor state estimation; ▪ integrate all modules into a coherent cognitive system architecture.		
Learning format	☒ Lecture		
	☐ Exercise		
	☐ Seminar/Seminar-style class		
	☒ Lab		
	☐ Project		
Recommended prerequisites	Competencies in accordance with the learning outcomes of the Bachelor's modules "Linear Algebra", "Probability Theory and Statistics" and "Machine Learning and Neural Networks"		
Reading list	▪ Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, 2005. ▪ Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge, 2004. ▪ Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, 2016. ▪ Siciliano, B. et al.: Robotics: Modelling, Planning and Control. Springer, 2010.		
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)		
	☒ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☐ Passing proficiency assessments		
Examination format	☒ Oral exam (if there are few participants)		
	☒ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☐ Project (with a presentation, if applicable)		
Eligibility	Computer Science		☐ Compulsory ☒ Elective
	Artificial Intelligence and Data Science		☐ Compulsory ☒ Elective
	Artificial Intelligence in Computer Science		☒ Compulsory ☐ Elective
Offer	☐ Summer term ☒ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Prof. Dr. Jürgen Graf		
Person in charge	Prof. Dr. Jürgen Graf		
Date of change	24.06.2026		

Cyber Security (E)			

Contents	▪ Introduction: Basic concepts and relationships
	▪ Cryptology
	  ▪ Quantum-resistant cryptographic schemes such as CRYSTALS Kyber
	  ▪ Fully homomorphic schemes such as BFV
	▪ Secure AI
	  ▪ Fundamentals and attacks on neural networks
	  ▪ Adversarial attacks
	▪ Organisational, conceptual and process-oriented security
	  ▪ Security concepts in accordance with the BSI's IT-Grundschutz
	  ▪ Security management and ISO 27000
	▪ Patch management
	▪ Where applicable, further topics relating to information security, such as the legal aspects of information security		

Learning outcomes	Students have gained a comprehensive understanding of information security and have come to understand it. This encompasses not only technical aspects but, in particular, organisational and conceptual aspects. In addition, current topics in cryptology and attacks on neural networks are covered in depth. Students are able to ▪ identify and apply structured approaches to implementing information security based on standards such as ISO 27x or BSI 100, ▪ carry out and calculate risk analyses, ▪ assess the impact of quantum computers on cryptology, ▪ explain the theory and functioning of the post-quantum methods covered, ▪ carry out and calculate the necessary operations of the post-quantum methods using small numerical examples, ▪ explain the taxonomy of attacks on neural networks, ▪ carry out the key steps of attacks on neural networks mathematically using simple examples, ▪ execute attacks on neural networks using provided code and adapt them to new datasets, describe the basic principles and standards of patch management and apply them in example exercises.		
Learning format	☐ Lecture		
	☒ Exercise		
	☒ Seminar/Seminar-style class		
	☒ Lab (Selected material and the exercises are explored in greater depth through practical exercises.)		
	☐ Project		
Recommended prerequisites	Competencies in accordance with the learning outcomes of the Bachelor's modules "IT Security"		
Reading list	▪ Ross Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, John Wiley & Sons ▪ Bundesamt für Sicherheit in der Informationstechnik, Standardreihe 200-x ▪ Claudia Eckert, IT-Sicherheit: Konzepte, Verfahren, Protokolle, Oldenbourg ▪ ISO 27001 und ISO 27002 ▪ Norbert Pohlmann: Cyber-Sicherheit, Springer, 2019 (Available as an eBook in the library)		
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)		
	☐ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☐ Passing proficiency assessments		
Examination format	☒ Oral exam (if there are few participants)		
	☒ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☐ Project (with a presentation, if applicable)		
Eligibility	Computer Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence and Data Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence in Computer Science	☐ Compulsory ☒ Elective	
Offer	☐ Summer term ☐ Winter term ☒ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Prof. Dr. Konstantin Knorr		

Person in charge	Prof. Dr. Konstantin Knorr
Date of change	29.06.2026

Data Science (D)			

Inhalte	▪ Grundlagen und Anwendungsfälle für Data Science
	▪ Daten
	  ▪ Typische Datenquellen im Unternehmen
	  ▪ Arten von Daten: Stammdaten, Bewegungsdaten
	  ▪ strukturierte, unstrukturierte und semistrukturierte Daten
	  ▪ Netzwerkdaten/Graphen
	  ▪ Werkzeuge zur Datenanalyse und -vorverarbeitung
	  ▪ Datenqualität
	▪ Datenverarbeitung
	  ▪ Architektur von Datenverarbeitungsstrecken
	  ▪ Umgang mit großen Datenmengen
	  ▪ Datenintegration
	  ▪ Data Ingestion
	  ▪ Präsentationsschicht
	▪ Business Intelligence
	  ▪ Grundlagen Data Warehousing
	  ▪ Reporting
	  ▪ Visualisierung
	▪ Empfehlungssysteme
	▪ Social Network Analysis		

Lernergebnisse	Die Studierenden sind nach der Teilnahme in der Lage ▪ vorhandene Datenbestände in Unternehmen zu bewerten, zu verarbeiten und zu nutzen, ▪ die Qualität und Struktur von Daten einzuschätzen, ▪ Daten aus verschiedenen Quellen aufzubereiten und zu integrieren, ▪ Daten und daraus abgeleitetes Wissen für Fachanwender zu präsentieren, ▪ gängige Werkzeuge wie Kafka, Camel, PDI, Pandas, Jupyter einzusetzen und ▪ Wissen und Empfehlungen aus sozialen Netzwerken zu generieren.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Bachelor-Module „Datenbanken“ und „Big-Data-Technologien“		
Literatur	▪ Cady, F.: The Data Science Handbook. Wiley, 2017. ▪ O’Neil, C., Schutt, R.: Doing Data Science. O’Reilly, 2014. ▪ Kotu, V., Deshpande, B.: Data Science: Concepts and Practice. 2. Auflage, Morgan Kaufmann, 2019. ▪ Barabasi, A.-L., Posfai, M.: Network Science. Cambridge University Press, 2016. ▪ Aggarwal, C. C.: Recommender Systems: The Textbook. Springer, 2016. ▪ Kimball, R., Ross, M.: The Data Warehouse Toolkit. 3. Auflage, Wiley, 2013.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik	☒ Pflicht ☐ Wahlpflicht	
	Künstliche Intelligenz und Data Science	☒ Pflicht ☐ Wahlpflicht	
	Artificial Intelligence in Computer Science	☒ Pflicht ☐ Wahlpflicht	
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Christoph Schmitz		

Modulverantwortliche(r)	Prof. Dr. Christoph Schmitz
Änderungsdatum	17.06.2026

Data Science (E)			

Contents	▪ Fundamentals and use cases for data science
	▪ Data
	  ▪ Typical data sources within an organisation
	  ▪ Types of data: master data, transaction data
	  ▪ Structured, unstructured and semi-structured data
	  ▪ Network data/graphs
	  ▪ Tools for data analysis and pre-processing
	  ▪ Data quality
	▪ Data processing
	  ▪ Architecture of data processing pipelines
	  ▪ Handling large volumes of data
	  ▪ Data integration
	  ▪ Data ingestion
	  ▪ Presentation layer
	▪ Business intelligence
	  ▪ Fundamentals of data warehousing
	  ▪ Reporting
	  ▪ Visualisation
	▪ Recommendation systems
	▪ Social network analysis		

Learning outcomes	Upon completion of the course, students will be able to ▪ evaluate, process and utilise existing data sets within organisations, ▪ assess the quality and structure of data, ▪ prepare and integrate data from various sources, ▪ present data and the knowledge derived from it to business users, ▪ use common tools such as Kafka, Camel, PDI, Pandas and Jupyter, and ▪ generate insights and recommendations from social networks.		
Learning format	☒ Lecture		
	☒ Exercise		
	☐ Seminar/Seminar-style class		
	☐ Lab		
	☐ Project		
Recommended prerequisites	Competencies in accordance with the learning outcomes of the Bachelor's modules "Databases" and "Big Data Technologies"		
Reading list	▪ Cady, F.: The Data Science Handbook. Wiley, 2017. ▪ O'Neil, C., Schutt, R.: Doing Data Science. O'Reilly, 2014. ▪ Kotu, V., Deshpande, B.: Data Science: Concepts and Practice. 2. Auflage, Morgan Kaufmann, 2019. ▪ Barabasi, A.-L., Posfai, M.: Network Science. Cambridge University Press, 2016. ▪ Aggarwal, C. C.: Recommender Systems: The Textbook. Springer, 2016. ▪ Kimball, R., Ross, M.: The Data Warehouse Toolkit. 3. Auflage, Wiley, 2013.		
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)		
	☐ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☐ Passing proficiency assessments		
Examination format	☒ Oral exam (if there are few participants)		
	☒ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☐ Project (with a presentation, if applicable)		
Eligibility	Computer Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence and Data Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence in Computer Science	☒ Compulsory ☐ Elective	
Offer	☒ Summer term ☐ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Prof. Dr. Christoph Schmitz		

Person in charge	Prof. Dr. Christoph Schmitz
Date of change	18.06.2026

Data Warehouse (D)			
Inhalte	▪ Motivation zum Einsatz von Data Warehouses (DWH) ▪ Einbettung von DWH in typische IT-Landschaften und -Prozesse ▪ SQL und PL/SQL ▪ Dimensionale Modellierung von DWH nach Kimball ▪ Implementierung von DWH in relationalen Datenbanksystemen ▪ OLAP und analytische Operatoren in SQL ▪ ETL-Prozesse (Extract-Transform-Load) und ETL-Tools ▪ Semistrukturierte Daten in XML und JSON		
Lernergebnisse	Die Studierenden können ▪ die Motivation für den Einsatz von DWH und typische Einsatzszenarien für DWH in IT-Landschaften erklären, ▪ geeignete Granularität, Dimensionen, Fakten und Metriken für gegebene Anwendungsszenarien von DWH auswählen, ▪ dimensionale Modelle nach Kimball in einer relationalen Datenbank implementieren, ▪ fortgeschrittene analytische Anfragen gegen relationale Datenbanken in SQL stellen, ▪ Daten mithilfe von PL/SQL, AWK und graphischen Werkzeugen wie z. B. PDI extrahieren und transformieren, sowie ▪ die Unterschiede zwischen semistrukturierten und strukturierten Daten verstehen und deren Verbindung innerhalb einer relationalen Datenbank umsetzen.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „Datenbanken“		
Literatur	▪ William Inmon: Building the Data Warehouse. 4. Auflage, Wiley, 2005. ▪ Ralph Kimball, Margy Ross: The Data Warehouse Toolkit. 3. Auflage, Wiley, 2013.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik	☐ Pflicht ☒ Wahlpflicht	
	Künstliche Intelligenz und Data Science	☒ Pflicht ☐ Wahlpflicht	
	Artificial Intelligence in Computer Science	☐ Pflicht ☒ Wahlpflicht	
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Christoph Schmitz		
Modulverantwortliche(r)	Prof. Dr. Christoph Schmitz		
Änderungsdatum	17.06.2026		

Ethical and Responsible AI (D)			
Inhalte	▪ Soziale und technische Aspekte von computergestützter Kooperation ▪ Klassifikation und Gestaltung von Groupware ▪ Automatisierung und Assistenz ▪ Ebenen der Mensch-Maschine-Kooperation ▪ Geteilte Kontrolle und Entscheidungsunterstützung ▪ Mensch-Maschine-Integration ▪ Ethische Grundsätze und gesellschaftliche Auswirkungen von KI-Systemen ▪ Verzerrungen (Bias) in Daten und Modellen des maschinellen Lernens ▪ Definitionen und Metriken für Fairness ▪ Bewertung von Modellen des maschinellen Lernens hinsichtlich Fairness und Risiko ▪ Risikobewertung und -management für KI-Systeme ▪ Rechtliche und regulatorische Rahmenbedingungen für KI-Systeme		
Lernergebnisse	Die Studierenden sind nach der Teilnahme in der Lage ▪ soziale und technische Bedingungen der Kooperation zwischen Menschen und KI-Systemen zu analysieren und deren Wechselwirkungen für unterschiedliche Anwendungskontexte herausarbeiten. ▪ Konzepte geteilter Kontrolle und algorithmischer Entscheidungsunterstützung auf konkrete Anwendungsszenarien anzuwenden und dabei Auswirkungen auf menschliche Autonomie, Verantwortungszuschreibung, Transparenz und Sicherheit zu bewerten. ▪ Gestaltungsalternativen für die Verteilung von Kontrolle und Entscheidungskompetenz zwischen Menschen und KI-Systemen zu entwickeln. ▪ Formen der Mensch-Maschine-Integration hinsichtlich ihrer technischen Voraussetzungen sowie ihrer individuellen und gesellschaftlichen Folgen kritisch beurteilen; ▪ ethische Grundprinzipien verantwortungsvoller KI-Systeme zu erläutern und auf den Einsatz von KI-Systemen in der Gesellschaft zu beziehen. ▪ Bias-Quellen zu identifizieren und zu unterscheiden und deren mögliche Auswirkungen auf Personen und Gruppen zu analysieren. ▪ KI-Systeme hinsichtlich Fairness, Risiko und potenzieller Schäden zu bewerten. ▪ Maßnahmen für verantwortungsvolle Entwicklung, Einführung und Überwachung von KI-Systemen abzuleiten.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ Barocas, S., Hardt, M., & Narayanan, A. (2023). Fairness and Machine Learning: Limitations and Opportunities. MIT Press. ▪ Hollnagel, E., & Woods, D. D. (2005). Joint cognitive systems. CRC Press. ▪ Sheridan, T. B. (2002). Humans and automation: System Design and research issues. John Wiley & Sons. ▪ Millot, P. (Ed.). (2014). Designing human-machine cooperation systems. Wiley-ISTE.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik		☐ Pflicht ☒ Wahlpflicht
	Künstliche Intelligenz und Data Science		☒ Pflicht ☐ Wahlpflicht
	Artificial Intelligence in Computer Science		☒ Pflicht ☐ Wahlpflicht
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium

	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Tilo Mentler, Prof. Dr. Claudia Schon		
Modulverantwortliche(r)	Prof. Dr. Tilo Mentler		
Änderungsdatum	30.06.2026		

Ethical and Responsible AI (E)			
Contents	▪ Social and technical aspects of computer-supported collaboration ▪ Classification and design of groupware ▪ Automation and assistance ▪ Levels of human-machine collaboration ▪ Shared control and decision support ▪ Human-machine integration ▪ Ethical principles and societal impacts of AI systems ▪ Bias in data and machine learning models ▪ Definitions and metrics of fairness ▪ Evaluation of machine learning models with regard to fairness and risk ▪ Risk assessment and management for AI systems ▪ Legal and regulatory frameworks for AI systems		
Learning outcomes	Upon completion of the course, students will be able to ▪ analyze the social and technical conditions of cooperation between humans and AI systems and identify their interactions in various application contexts. ▪ apply concepts of shared control and algorithmic decision support to specific application scenarios, while assessing their impact on human autonomy, attribution of responsibility, transparency, and security. ▪ develop design alternatives for the distribution of control and decision-making authority between humans and AI systems. ▪ critically evaluate forms of human-machine integration in terms of their technical requirements as well as their individual and societal consequences ▪ explain ethical principles of responsible AI systems and relate them to the use of AI systems in society. ▪ identify and distinguish sources of bias and analyse their potential impacts on individuals and groups. ▪ evaluate AI systems with regard to fairness, risk and potential harm. ▪ derive measures for the responsible development, deployment and monitoring of AI systems.		
Learning format	☒ Lecture		
	☒ Exercise		
	☐ Seminar/Seminar-style class		
	☐ Lab		
	☐ Project		
Recommended prerequisites	None		
Reading list	▪ Barocas, S., Hardt, M., & Narayanan, A. (2023). Fairness and Machine Learning: Limitations and Opportunities. MIT Press. ▪ Hollnagel, E., & Woods, D. D. (2005). Joint cognitive systems. CRC Press. ▪ Sheridan, T. B. (2002). Humans and automation: System Design and research issues. John Wiley & Sons. ▪ Millot, P. (Ed.). (2014). Designing human-machine cooperation systems. Wiley-ISTE.		
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)		
	☐ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☐ Passing proficiency assessments		
Examination format	☐ Oral exam		
	☐ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☒ Project (with a presentation, if applicable)		
Eligibility	Computer Science	☐ Compulsory ☒ Elective	
	Artificial Intelligence and Data Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence in Computer Science	☒ Compulsory ☐ Elective	
Offer	☐ Summer term ☒ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours

Lecturer	Prof. Dr. Tilo Mentler, Prof. Dr. Claudia Schon
Person in charge	Prof. Dr. Tilo Mentler
Date of change	30.06.2026

Fortgeschrittene Mensch-Computer-Interaktion (D)			
Inhalte	▪ Grundbegriffe und fortgeschrittene Konzepte der Mensch-Computer-Interaktion ▪ Menschliche Kognition, Emotion, Motivation und Handlungsprozesse ▪ 6 Perspektiven und Ebenen der Interaktion ▪ Mentale Modelle und Metaphern ▪ Analyse von Nutzungskontexten ▪ Nutzungsanforderungen und Gestaltungslösungen ▪ Formative und summative Evaluationsverfahren ▪ Qualitative und Quantitative Forschung ▪ Publizieren und Präsentieren im Zusammenhang mit Mensch-Computer-Interaktion		
Lernergebnisse	Die Studierenden können ▪ Erkenntnisse über menschliche Kognition, Emotion, Motivation und Handlungsprozesse analysieren und daraus begründete Gestaltungsentscheidungen für interaktive Systeme ableiten. ▪ für konkrete Nutzungskontexte geeignete Analyse-, Gestaltungs- und Evaluationsverfahren auswählen, auf Benutzungsschnittstellen anwenden und ihre Auswahl methodisch begründen. ▪ Forschungsergebnisse und Gestaltungsalternativen der Mensch-Computer-Interaktion anhand wissenschaftlicher Kriterien hinsichtlich ihrer methodischen Qualität und Aussagekraft beurteilen. ▪ Prototypen, empirische Studien und wissenschaftliche Arbeiten kriteriengeleitet analysieren und begründetes, adressatengerechtes Feedback formulieren. ▪ zu einer komplexen Fragestellung der Mensch-Computer-Interaktion eigenständig ein methodisch begründetes Forschungsdesign entwickeln.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ Hornbæk, K., Kristensson, P. O., & Oulasvirta, A. (2025). Introduction to Human-Computer Interaction. Oxford University Press. ▪ Heinecke, A. M., & Gerken, J. (2024). Mensch-Computer-Interaktion: Basiswissen für Entwickler und Gestalter ▪ Rogers Y., Sharp H. & Preece J. (2023). Interaction design: beyond human-computer interaction. Hoboken, New Jersey: John Wiley & Sons, Inc. ▪ Reuter, C. (Hrsg.) (2018). Sicherheitskritische Mensch-Computer-Interaktion. Wiesbaden: Springer Fachmedien Wiesbaden. ▪ Herczeg, M. (2009). Software-Ergonomie: Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme. (3 Aufl.) (Interaktive Medien). De Gruyter Studium. ▪ Dahm, M. (2006). Grundlagen der Mensch-Computer-Interaktion. Informatik: Software-Ergonomie. München: Pearson Studium.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik	☒ Pflicht ☐ Wahlpflicht	
	Künstliche Intelligenz und Data Science	☐ Pflicht ☒ Wahlpflicht	
	Artificial Intelligence in Computer Science	☐ Pflicht ☒ Wahlpflicht	
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Tilo Mentler		
Modulverantwortliche(r)	Prof. Dr. Tilo Mentler		

Änderungsdatum	29.06.2026

Fortgeschrittene Programmierkonzepte (D)	
Inhalte	Dieses Modul behandelt fortgeschrittene Programmierkonzepte und deren Anwendung in modernen Programmiersprachen. Aufbauend auf bereits vorhandenen Kenntnissen in Programmierung und Softwareentwicklung werden unter anderem folgende Themen behandelt: ▪ Abstraktion ▪ Fortgeschrittene Typsysteme ▪ Funktionale Programmierung ▪ Nebenläufigkeitsmodelle ▪ Metaprogrammierung ▪ Speicherverwaltung ▪ Laufzeitsysteme von Programmiersprachen ▪ Fortgeschrittene Datenstrukturen ▪ Algorithmische Entwurfs- und Entwicklungstechniken Die Studierenden entwickeln ein vertieftes Verständnis der grundlegenden Prinzipien moderner Programmiersprachen und lernen, Programmierparadigmen, Sprachmerkmale, Entwurfsansätze sowie algorithmische Lösungsstrategien systematisch zu analysieren und zu vergleichen. Das Modul verbindet theoretische Grundlagen mit praktischen Programmierbeispielen und Übungen. Dazu gehören die Implementierung und Evaluation von Algorithmen und Datenstrukturen auf großen Datenmengen. Python dient dabei als primäre Implementierungssprache. Die behandelten Konzepte werden jedoch sprachunabhängig vermittelt und mit Sprachmerkmalen zahlreicher moderner Programmiersprachen in Beziehung gesetzt.
Lernergebnisse	Die Studierenden sind nach der Teilnahme in der Lage ▪ fortgeschrittene Programmierkonzepte und Sprachmerkmale zu erläutern und in praktischen Programmieraufgaben anzuwenden, ▪ Programmierparadigmen, Sprachdesigns, Implementierungsansätze sowie algorithmische Lösungsstrategien zu analysieren und zu vergleichen, ▪ fortgeschrittene Datenstrukturen und Algorithmen für komplexe Programmierprobleme zu entwerfen, zu implementieren und zu evaluieren, ▪ das Verhalten, die Korrektheit, die Leistungsfähigkeit, die Skalierbarkeit, die Nebenläufigkeit sowie die Speicherverwaltung von Programmen anhand geeigneter konzeptioneller Modelle zu analysieren und zu beurteilen, ▪ Zielkonflikte und Abwägungen zwischen alternativen Programmiertechniken, Datenstrukturen, Algorithmen und Sprachmerkmalen zu bewerten, ▪ fortgeschrittene Programmierkonzepte auf unbekannte Programmiersprachen, Frameworks und Technologien zu übertragen, ▪ programmierbezogene Konzepte, algorithmische Entwurfsentscheidungen sowie Implementierungsalternativen und deren Konsequenzen unter Verwendung präziser technischer Fachterminologie zu kommunizieren und zu diskutieren.
Lehrform	☒ Vorlesung
	☒ Übung
	☐ Seminar/Seminaristischer Unterricht
	☒ Labor
	☐ Projekt
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Bachelor-Module „Einführung in die Programmierung", „Objektorientierte Programmierung - Grundlagen", „Objektorientierte Programmierung - Vertiefung", „Software-Entwurf" und „Datenstrukturen und Algorithmen" sowie grundlegende Kenntnisse der Komplexitätstheorie, insbesondere der Laufzeitanalyse und grundlegender Komplexitätsklassen.
Literatur	▪ Ramalho, L.: Fluent Python: Clear, Concise, and Effective Programming (2nd ed.). O'Reilly Media, 2022. ▪ Viafore, P.: Robust Python: Write Clean and Maintainable Code (1st ed.). O'Reilly Media, 2021. ▪ Van Roy, P., & Haridi, S.: Concepts, Techniques, and Models of Computer Programming (1st ed.). MIT Press, 2004. ▪ Scott, M. L., & Aldrich, J.: Programming Language Pragmatics (5th ed.). Morgan Kaufmann / Elsevier, 2025. ▪ Cormen et al: Introduction to Algorithms (3rd Edition), 2009, The MIT Press ▪ Hetland: Python Algorithms, 2010, APress
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)
	☒ Bestehen von Leistungsstandkontrollen
Prüfungsform	☐ Mündliche Prüfung
	☒ Klausur
	☐ Prüfung am PC
	☐ Hausarbeit (ggf. mit Präsentation)
	☐ Projekt (ggf. mit Präsentation)

Verwendbarkeit	Informatik		☒ Pflicht ☐ Wahlpflicht
	Künstliche Intelligenz und Data Science		☒ Pflicht ☐ Wahlpflicht
	Artificial Intelligence in Computer Science		☒ Pflicht ☐ Wahlpflicht
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Thomas Hartmann, Prof. Dr. Heinz Schmitz		
Modulverantwortliche(r)	Prof. Dr. Thomas Hartmann		
Änderungsdatum	07.07.2026		

Fortgeschrittene Symbolische KI (D)			

Inhalte	▪ Wissensrepräsentation und automatisches Schließen
	  ▪ Formale Modellierung von Wissen, Regeln und Beziehungen
	  ▪ Automatische Ableitung impliziten Wissens aus expliziten Wissensbasen
	▪ SAT- und SMT-basierte Modellierung und Problemlösung
	  ▪ Solver-basierte Ansätze für Entscheidungs-, Planungs- und Konfigurationsprobleme
	▪ Nichtmonotones Schließen
	  ▪ Schließen mit unvollständigem, veränderlichem oder widersprüchlichem Wissen
	  ▪ Answer Set Programming
	▪ Knowledge Graphs als strukturierte Wissensbasis für symbolisches Schließen
	▪ Erklärungen und Begründungen für symbolische Schlussfolgern
	▪ Kombination symbolischer und datengetriebener Verfahren, insbesondere neuro-symbolisches Schließen		

Lernergebnisse	Die Studierenden sind nach der Teilnahme in der Lage ▪ Ansätze der Wissensrepräsentation zu erläutern und Wissen, Regeln und Beziehungen formal zu modellieren. ▪ Verfahren des automatischen Schließens anzuwenden, um implizites Wissen aus expliziten Wissensbasen abzuleiten. ▪ SAT- und SMT-basierte Ansätze zur Modellierung und Lösung von Entscheidungs-, Planungs- und Konfigurationsproblemen einzuordnen und auf ausgewählte Problemstellungen anzuwenden. ▪ nichtmonotone Schlussfolgerungsverfahren zu erklären und für den Umgang mit unvollständigem, veränderlichem oder widersprüchlichem Wissen zu nutzen. ▪ Knowledge Graphs als strukturierte Wissensbasis für symbolisches Schließen zu beschreiben und grundlegende Anfrage- und Inferenzmöglichkeiten einzuordnen. ▪ Erklärungen und Begründungen für symbolische Schlussfolgerungen nachzuvollziehen und deren Bedeutung für hybride und neuro-symbolische KI-Systeme zu reflektieren.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ Stuart Russell, Peter Norvig. Artificial Intelligence: A Modern Approach. Fourth Edition, Pearson, 2020 ▪ Barrett, C. et al. (2025). Satisfiability Modulo Theories: A Beginner's Tutorial. In: Platzer, A., Rozier, K.Y., Pradella, M., Rossi, M. (eds) Formal Methods. FM 2024. Lecture Notes in Computer Science, vol 14934. Springer, Cham ▪ Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Torsten Schaub: Answer Set Solving in Practice. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers 2012		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☒ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik		☐ Pflicht ☒ Wahlpflicht
	Künstliche Intelligenz und Data Science		☒ Pflicht ☐ Wahlpflicht
	Artificial Intelligence in Computer Science		☒ Pflicht ☐ Wahlpflicht
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Claudia Schon		
Modulverantwortliche(r)	Prof. Dr. Claudia Schon		
Änderungsdatum	29.06.2026		

Generative AI (E)			
Contents	▪ Fundamentals of Mathematics and Machine Learning: Probability Theory, Neural Networks ▪ Fundamentals of Generative Models: Latent Variables, VAEs and GANs, Diffusion Models ▪ Foundation Models and Transformers ▪ Data, Training, and Scaling: Data Curation, Large-Scale Training, Efficient Architectures ▪ Adaptation and Post-Training: Fine-Tuning, Compression ▪ Knowledge-Augmented AI Systems ▪ Multimodal Generative AI: Vision-Language Models, Cross-Modal Learning ▪ Evaluation and Reliability: Benchmarks, Hallucinations, Robustness ▪ Security: Adversarial Attacks, Jailbreak		
Learning outcomes	Students have acquired advanced knowledge of generative artificial intelligence and foundation models, including their theoretical foundations, training methodologies, evaluation, and practical deployment. They have developed the ability to analyze complex AI systems, select and apply appropriate methods to solve real-world problems, and critically assess model capabilities, limitations, and risks. Students are further enabled to independently study and evaluate current scientific literature, particularly in English, and to recognize ethical, societal, and security-related challenges associated with modern AI technologies.		
Learning format	☒ Lecture		
	☒ Exercise		
	☐ Seminar/Seminar-style class		
	☐ Lab		
	☐ Project		
Recommended prerequisites	None		
Reading list	▪ I. Goodfellow et al.: Deep Learning. MIT Press, 2016. ▪ C. Bishop, H. Bishop: Deep Learning. Springer 2024.		
Examination prerequisite	☐ Regular completion of homework assignments (with a presentation, if applicable)		
	☒ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☐ Passing proficiency assessments		
Examination format	☒ Oral exam		
	☐ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☐ Project (with a presentation, if applicable)		
Eligibility	Computer Science	☐ Compulsory ☒ Elective	
	Artificial Intelligence and Data Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence in Computer Science	☒ Compulsory ☐ Elective	
Offer	☒ Summer term ☐ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Prof. Dr. Hans-Peter Beise		
Person in charge	Prof. Dr. Hans-Peter Beise		
Date of change	19.06.2026		

Generative KI (D)			
Inhalte	▪ Mathematische und Machine Learning-Grundlagen: Wahrscheinlichkeitstheorie, Neuronale Netze ▪ Grundlagen generativer Modelle: Latent Variables, VAEs und GANs, Diffusion Models ▪ Foundation Models und Transformer ▪ Daten, Training und Skalierung: Datenaufbereitung, Large-Scale Training, Effiziente Architekturen ▪ Adaptation und Post-Training: Fine-Tuning, Kompression ▪ Knowledge-Augmented AI Systems ▪ Multimodale Generative KI: Vision-Language Models, Cross-Modal Learning ▪ Evaluation und Zuverlässigkeit: Benchmarks, Halluzinationen, Robustheit ▪ Sicherheit: Adversarial Attacks, Jailbreak		
Lernergebnisse	Die Studierenden haben vertiefte Kenntnisse im Bereich der Generativen Künstlichen Intelligenz und von Foundation Models (wie LLM) erworben. Sie haben gelernt, komplexe Problemstellungen zu analysieren, geeignete KI-Methoden auszuwählen und anzuwenden sowie die Leistungsfähigkeit, Zuverlässigkeit und Grenzen von Modellen kritisch zu bewerten. Die Studierenden sind in der Lage, sich selbstständig aktuelles Wissen aus wissenschaftlicher Primärliteratur anzueignen und ethische, gesellschaftliche sowie sicherheitsrelevante Aspekte moderner KI-Systeme kritisch einzuordnen.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ I. Goodfellow et al.: Deep Learning. MIT Press, 2016. ▪ C. Bishop, H. Bishop: Deep Learning. Springer 2024.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik	☐ Pflicht ☒ Wahlpflicht	
	Künstliche Intelligenz und Data Science	☒ Pflicht ☐ Wahlpflicht	
	Artificial Intelligence in Computer Science	☒ Pflicht ☐ Wahlpflicht	
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Hans-Peter Beise		
Modulverantwortliche(r)	Prof. Dr. Hans-Peter Beise		
Änderungsdatum	19.06.2026		

<table>
<tr><td colspan="5">German Language and Culture (E)</td></tr>
<tr><td>Contents</td><td colspan="4">▪ German language foundations for everyday university life: pronunciation, spelling, numbers, time, dates, classroom language and basic interaction patterns.
▪ Basic grammar and vocabulary: articles, nouns, plural forms, personal pronouns, present tense of regular and common irregular verbs, modal verbs, negation, separable verbs, word order in main clauses and questions.
▪ Communicative situations: introducing oneself, talking about studies and origin, asking for information, orientation on campus and in town, shopping, food, housing, appointments and simple e-mail messages.
▪ Intercultural orientation: university culture in Germany, forms of address, politeness conventions, everyday routines, public services, student support structures and regional/cultural aspects of life in Germany.
▪ Integrated practice in listening, speaking, reading and writing with short authentic texts and simple digital learning activities.</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">Upon completion of the course, students will be able to
▪ understand and use familiar everyday expressions and basic phrases in academic and social situations at university.
▪ introduce themselves and others, ask and answer simple questions about personal background, studies, schedules, places and needs.
▪ read and write short, simple texts such as messages, forms, short e-mails and notices related to student life.
▪ apply basic German grammar and vocabulary in spoken and written communication with increasing accuracy.
▪ recognise important cultural conventions in German higher education and everyday life and act appropriately in routine situations.
▪ progress from A1 towards A2 of the Common European Framework of Reference (CEFR), depending on entry level and individual workload.</td></tr>
<tr><td rowspan="5">Learning format</td><td colspan="4">☒ Lecture</td></tr>
<tr><td colspan="4">☒ Exercise</td></tr>
<tr><td colspan="4">☒ Seminar/Seminar-style class</td></tr>
<tr><td colspan="4">☐ Lab</td></tr>
<tr><td colspan="4">☐ Project</td></tr>
<tr><td>Recommended prerequisites</td><td colspan="4">None; intended for non-native speakers with no or very limited prior knowledge of German (target entry level: A1 or below).</td></tr>
<tr><td>Reading list</td><td colspan="4">▪ Netzwerk neu A1/A2 or equivalent current CEFR-oriented German course material
▪ Schritte international neu A1/A2 or comparable textbook/workbook materials
▪ Digital exercises and self-study materials provided via the university learning platform
▪ Selected short authentic texts from university and everyday contexts</td></tr>
<tr><td rowspan="3">Examination prerequisite</td><td colspan="4">☒ Regular completion of homework assignments (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☐ Completion of term paper/lab assignments (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☒ Passing proficiency assessments</td></tr>
<tr><td rowspan="5">Examination format</td><td colspan="4">☒ Oral exam</td></tr>
<tr><td colspan="4">☐ Written exam</td></tr>
<tr><td colspan="4">☐ Computer-based exam</td></tr>
<tr><td colspan="4">☐ Term paper (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☐ Project (with a presentation, if applicable)</td></tr>
<tr><td rowspan="3">Eligibility</td><td colspan="3">Computer Science</td><td>☐ Compulsory ☐ Elective</td></tr>
<tr><td colspan="3">Artificial Intelligence and Data Science</td><td>☐ Compulsory ☐ Elective</td></tr>
<tr><td colspan="3">Artificial Intelligence in Computer Science</td><td>☒ Compulsory ☐ Elective (non-native speakers only)</td></tr>
<tr><td>Offer</td><td colspan="4">☐ Summer term ☒ Winter term ☐ Irregular</td></tr>
<tr><td rowspan="2">Workload</td><td>ECTS credits</td><td>Contact time</td><td colspan="2">Self-study</td></tr>
<tr><td>5</td><td>60 hours</td><td colspan="2">90 hours</td></tr>
<tr><td>Lecturer</td><td colspan="4">Prof. Dr. Georg Schneider</td></tr>
<tr><td>Person in charge</td><td colspan="4">Prof. Dr. Georg Schneider</td></tr>
<tr><td>Date of change</td><td colspan="4">08.07.2026</td></tr>
</table>

<table>
<tr><td colspan="5">High Performance Computing (E)</td></tr>
<tr><td>Contents</td><td colspan="4">▪ General paradigms for parallel programming
▪ Using the Threading Building Blocks Library for CPU parallelism
  ▪ Concepts of the task stealing scheduler
  ▪ Efficient memory management for parallel systems
  ▪ General parallelism concepts in TBB
▪ Using CUDA for GPGPU computing
  ▪ Concepts of GPGPU computing
  ▪ Writing simple CUDA programs
  ▪ Synchronization in CUDA
  ▪ Streaming and overlapping in CUDA</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">Students are enabled to decide which HPC concept to use for a specific task at hand for compute intensive operations. They are enabled to implement those in concrete programs.</td></tr>
<tr><td rowspan="5">Learning format</td><td colspan="4">☒ Lecture</td></tr>
<tr><td colspan="4">☒ Exercise</td></tr>
<tr><td colspan="4">☐ Seminar/Seminar-style class</td></tr>
<tr><td colspan="4">☐ Lab</td></tr>
<tr><td colspan="4">☐ Project</td></tr>
<tr><td>Recommended prerequisites</td><td colspan="4">Competencies in accordance with the learning outcomes of the Bachelor’s modules "C/C++ Programming" and "Computer Engineering"</td></tr>
<tr><td>Reading list</td><td colspan="4">▪ Michael McCool, Arch D. Robinson, James Reinders: Structured Parallel Programming. Morgan Kaufmann, 2012.
▪ James Reinders: Intel Threading Building Blocks. O’Reilly, 2007.
▪ Jason Sanders, Edward Kandrot: CUDA by Example. Addison-Wesley Professional, 2010.</td></tr>
<tr><td rowspan="3">Examination prerequisite</td><td colspan="4">☐ Regular completion of homework assignments (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☒ Completion of term paper/lab assignments (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☐ Passing proficiency assessments</td></tr>
<tr><td rowspan="5">Examination format</td><td colspan="4">☐ Oral exam</td></tr>
<tr><td colspan="4">☐ Written exam</td></tr>
<tr><td colspan="4">☐ Computer-based exam</td></tr>
<tr><td colspan="4">☐ Term paper (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☒ Project (with a presentation, if applicable)</td></tr>
<tr><td rowspan="3">Eligibility</td><td colspan="3">Computer Science</td><td>☐ Compulsory ☒ Elective</td></tr>
<tr><td colspan="3">Artificial Intelligence and Data Science</td><td>☐ Compulsory ☒ Elective</td></tr>
<tr><td colspan="3">Artificial Intelligence in Computer Science</td><td>☐ Compulsory ☒ Elective</td></tr>
<tr><td>Offer</td><td colspan="4">☐ Summer term ☐ Winter term ☒ Irregular</td></tr>
<tr><td rowspan="2">Workload</td><td>ECTS credits</td><td>Contact time</td><td colspan="2">Self-study</td></tr>
<tr><td>5</td><td>60 hours</td><td colspan="2">90 hours</td></tr>
<tr><td>Lecturer</td><td colspan="4">Prof. Dr. Christoph Lürig</td></tr>
<tr><td>Person in charge</td><td colspan="4">Prof. Dr. Christoph Lürig</td></tr>
<tr><td>Date of change</td><td colspan="4">18.06.2026</td></tr>
</table>

Informationssicherheit (D)			

Inhalte	▪ Einführung: Grundlegende Begriffe und Zusammenhänge
	▪ Kryptologie
	  ▪ Quanten-resistente kryptologische Verfahren wie CRYSTALS Kyber
	  ▪ Voll homomorphe Verfahren wie z.B. BFV
	▪ Secure AI
	  ▪ Grundlagen und Angriffe auf neuronale Netze
	  ▪ Adversarial Attacks
	▪ Organisatorische, konzeptionelle und prozessorientierte Sicherheit
	  ▪ Sicherheitskonzepte gemäß IT-Grundschutz des BSI
	  ▪ Sicherheitsmanagement und ISO 27000
	▪ Patch Management
	▪ Ggf. weitere Themen der Informationssicherheit wie z.B. rechtliche Aspekte der Informationssicherheit		

Lernergebnisse	Die Studierenden haben einen gesamtheitlichen Einblick in die Informationssicherheit erlernt und verstanden. Dies umfasst neben technischen insbesondere organisatorische und konzeptionelle Aspekte. Zusätzlich werden vertiefend aktuelle Themen der Kryptologie und Angriffe auf neuronale Netze behandelt. Die Studierenden können ▪ strukturierte Vorgehensweisen zur Umsetzung von Informationssicherheit anhand von Standards wie z.B. ISO 27x oder BSI 100 benennen und anwenden, ▪ Risikoanalysen durchführen und berechnen, ▪ die Auswirkungen von Quantencomputern auf die Kryptologie einschätzen, ▪ die Theorie und Funktionsweise der behandelten Post-Quanten-Verfahren erklären, ▪ in kleinen Zahlenbeispielen die notwendigen Operationen der Post-Quanten-Verfahren durchführen und berechnen, ▪ die Taxonomie der Angriffe auf neuronale Netzwerke erklären, ▪ die wichtigsten Schritte der Angriffe auf neuronale Netzwerke in kleinen Beispielen rechnerisch durchführen, ▪ Angriffe auf neuronale Netze per vorgegebenem Code ausführen und auf neue Datensätze übertragen, ▪ die Grundprinzipien und Standards des Patch Managements beschreiben und in Beispielaufgaben anwenden.		
Lehrform	☐ Vorlesung		
	☒ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☒ Labor (Ausgewählter Vorlesungsstoff und die Übungen werden vertieft durch praktische Übungen.)		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „IT-Sicherheit“		
Literatur	▪ Ross Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, John Wiley & Sons ▪ Bundesamt für Sicherheit in der Informationstechnik, Standardreihe 200-x ▪ Claudia Eckert, IT-Sicherheit: Konzepte, Verfahren, Protokolle, Oldenbourg ▪ ISO 27001 und ISO 27002 ▪ Norbert Pohlmann: Cyber-Sicherheit, Springer, 2019 (Als eBook in der Bibliothek vorhanden)		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik	☒ Pflicht ☐ Wahlpflicht	
	Künstliche Intelligenz und Data Science	☒ Pflicht ☐ Wahlpflicht	
	Artificial Intelligence in Computer Science	☐ Pflicht ☒ Wahlpflicht	
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden

Lehrende(r)	Prof. Dr. Konstantin Knorr
Modulverantwortliche(r)	Prof. Dr. Konstantin Knorr
Änderungsdatum	29.06.2026

Interactive Physical Simulation (E)			

Contents	▪ Collision detection
	  ▪ Coarse granular collision detection (sweep and prune, BSP trees, Octrees)
	  ▪ Fine granular collision detection (primitive collision, hierarchical OBBS, Minkowski sums, k-dops)
	  ▪ Tunnel Problem and Solutions like conservative advancement and retroactive detection
	▪ physical properties of moving rigid bodies (force, torque, acceleration, momentum, impulse, velocity...), Newton mechanic
	▪ numerical Integration of ODEs, L- and A- stability, implicit semi implicit and explicit solvers.
	▪ Collision reaction
	  ▪ Collision reaction between two rigid bodies with one point of contact
	  ▪ Multiple rigid bodies and multiple points of contact
	  ▪ Impulse base Simulation methods
	▪ Contraint based Simulation methods
	  ▪ Early methods like penalty and verlet integration
	  ▪ Lagrange Dynamics for contrained motion
	  ▪ linear complimentary problems
	  ▪ Simulation of breaking and tearing for games with lagrange dynamics
	  ▪ Coupling of Simulation and Animation for articulated characters		

Learning outcomes	Students are enabled to understand the techniques used in physical simulators for games, apply the proper technique for the specific situation at hand and implement portions of the system themselves.		
Learning format	☒ Lecture		
	☒ Exercise		
	☐ Seminar/Seminar-style class		
	☐ Lab		
	☐ Project		
Recommended prerequisites	Competencies in accordance with the learning outcomes of the Bachelor's modules "Linear Algebra" and "Applied Mathematics"		
Reading list	▪ H. R. Schwarz: Numerische Mathematik. Springer Vieweg, 2011. ▪ David H. Eberly: Game Physics. Taylor & Francis, 2nd edition, 2010. ▪ Gino van den Bergen: Collision Detection in Interactive 3D Environments. CRC Press, 2003.		
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)		
	☐ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☐ Passing proficiency assessments		
Examination format	☒ Oral exam		
	☐ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☐ Project (with a presentation, if applicable)		
Eligibility	Computer Science	☐ Compulsory ☒ Elective	
	Artificial Intelligence and Data Science	☐ Compulsory ☒ Elective	
	Artificial Intelligence in Computer Science	☐ Compulsory ☒ Elective	
Offer	☐ Summer term ☐ Winter term ☒ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Prof. Dr. Christoph Lürig		
Person in charge	Prof. Dr. Christoph Lürig		
Date of change	18.06.2026		

Knowledge Representation with Description Logics (E)			
Contents	Knowledge representation deals with the representation of knowledge in a form that can be processed by information systems. Description logics have a long tradition in computer science and knowledge representation. They were developed to model domain knowledge and to enable systems to draw logical conclusions about that knowledge. In recent years, description logics have gained in importance as they form the logical basis for widely used ontology languages. This lecture will examine various description logics and explore their properties in more detail. Topics covered include: ▪ An overview of the field of knowledge representation ▪ The description logic ALC ▪ Inference problems ▪ Expressiveness and properties of the models ▪ Extensions of ALC ▪ Tableau algorithms ▪ Justifications for inferences ▪ The description logic EL ▪ Query answering		
Learning outcomes	Students are enabled to ▪ represent knowledge in the form of description logic ontologies, ▪ design ontologies in Protégé, ▪ draw conclusions from the represented knowledge, ▪ use reasoners to draw conclusions from ontologies, ▪ generate explanations for conclusions.		
Learning format	☒ Lecture		
	☒ Exercise		
	☐ Seminar/Seminar-style class		
	☐ Lab		
	☐ Project		
Recommended prerequisites	None		
Reading list	▪ Baader, F., Horrocks I. Lutz C., Sattler U. An Introduction to Description Logic. Cambridge University Press; 2017 ▪ Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. (Eds.). (2007). The Description Logic Handbook: Theory, Implementation and Applications (2nd ed.). Cambridge: Cambridge University Press.		
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)		
	☐ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☐ Passing proficiency assessments		
Examination format	☒ Oral exam (if there are few participants)		
	☒ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☐ Project (with a presentation, if applicable)		
Eligibility	Computer Science		☐ Compulsory ☒ Elective
	Artificial Intelligence and Data Science		☐ Compulsory ☒ Elective
	Artificial Intelligence in Computer Science		☐ Compulsory ☒ Elective
Offer	☐ Summer term ☐ Winter term ☒ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Prof. Dr. Claudia Schon		
Person in charge	Prof. Dr. Claudia Schon		
Date of change	22.06.2026		

Learning Analytics and Educational Data Mining (D)				
Inhalte	Das Modul präsentiert und diskutiert die Analyse von Daten aus Lehr- und Lernsystemen als vielseitiges Anwendungsfeld von Data Science und KI-Methoden. Die Lerninhalte umfassen sowohl theoretische und praktische Grundlagen (z. B. Modelle, Architekturen, Rechtliche und Ethische Aspekte) als auch konkrete Anwendungsbeispiele durch die intensive Auseinandersetzung mit ausgewählten Verfahren und Algorithmen anhand realer Datensätze.			
Lernergebnisse	Die Studierenden können ▪ Konzepte und Anwendungsfelder von Learning Analytics und Educational Data Mining beschreiben, ▪ das „Learning Analytics Reference Model" sowie gängige Datenmodelle und Architekturen erklären, ▪ Verfahren und Algorithmen passend zu einem gegebenen Datensatz und einem gegebenen Analyseziel auswählen, ▪ ausgewählte Verfahren und Algorithmen auf konkrete Datensätze anwenden und die dadurch gewonnenen Erkenntnisse erklären, bewerten, präsentieren und diskutieren, ▪ Visualisierungen und Dashboards zur Unterstützung von Learning Analytics und Educational Data Mining entwerfen, ▪ Verfahren und Datensätze hinsichtlich rechtlicher und ethischer Bedenken einschätzen.			
Lehrform	☒ Vorlesung			
	☒ Übung			
	☐ Seminar/Seminaristischer Unterricht			
	☐ Labor			
	☐ Projekt			
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Module „Statistics and Learning Theory", „Data Science" und „Maschinelles Lernen und Neuronale Netze"			
Literatur	▪ Lang, Charles, et al., eds. "Handbook of learning analytics." (2017). ▪ Clow, Doug. "An overview of learning analytics." Teaching in Higher Education 18.6 (2013): 683-695. ▪ Chatti, Mohamed Amine, et al. "A reference model for learning analytics." International Journal of Technology-Enhanced Learning 4.5-6 (2012): 318-331. ▪ Romero, Cristobal, and Sebastian Ventura. "Educational data mining and learning analytics: An updated survey." Wiley interdisciplinary reviews: Data mining and knowledge discovery 10.3 (2020): e1355.			
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)			
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)			
	☐ Bestehen von Leistungsstandkontrollen			
Prüfungsform	☐ Mündliche Prüfung			
	☐ Klausur			
	☐ Prüfung am PC			
	☐ Hausarbeit (ggf. mit Präsentation)			
	☒ Projekt (ggf. mit Präsentation)			
Verwendbarkeit	Informatik		☐ Pflicht ☒ Wahlpflicht	
	Künstliche Intelligenz und Data Science		☐ Pflicht ☒ Wahlpflicht	
	Artificial Intelligence in Computer Science		☐ Pflicht ☒ Wahlpflicht	
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig			
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium	
	5	60 Stunden	90 Stunden	
Lehrende(r)	Prof. Dr. Michael Striewe			
Modulverantwortliche(r)	Prof. Dr. Michael Striewe			
Änderungsdatum	29.06.2026			

Machine Learning and Neural Networks (E)			
Contents	Machine learning methods form a central foundation for intelligent systems and numerous industrial applications. The focus is on training data-driven models as well as on the analysis, preprocessing, and dimensionality reduction of data. The methods covered are used to solve classification, regression, and clustering tasks; dimensionality reduction methods also support the visualization and exploratory analysis of complex data structures. In the field of machine learning, fundamental concepts are covered, including data preprocessing and dimensionality reduction, unsupervised learning methods such as k-means, PCA, MDS, Sammon mapping, and t-SNE, as well as supervised learning methods such as k-nearest neighbour, Bayes classifiers, decision trees, linear discriminant analysis, and support vector machines. In addition, linearly and non-linearly separable problems, quadratic error minimization, gradient descent methods, and the evaluation of learning methods are addressed. In the field of neural networks, fundamental architectures, learning principles, and training methods are covered. These include the structure of artificial neurons, activation functions, feedforward networks, backpropagation, loss functions, regularization, and optimization methods. Furthermore, key network architectures such as fully connected networks, convolutional neural networks, and transformer models are introduced and examined with regard to their areas of application, training requirements, and performance.		
Learning outcomes	The students are familiar with fundamental concepts, key methods, and the current state of research in the fields of machine learning and neural networks. They are able to understand, analyse, and critically reflect on current developments. Furthermore, they can independently implement selected methods, apply them to suitable data, evaluate their results, and assess the performance of the methods used.		
Learning format	☒ Lecture		
	☐ Exercise		
	☐ Seminar/Seminar-style class		
	☐ Lab		
	☐ Project		
Recommended prerequisites	None		
Reading list	▪ Duda, Richard O.: Pattern Classification: Pattern Classification. Wiley-Interscience. ▪ Peter Flach: Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press. ▪ Ferdinand van der Heijden, de Ridder, Dick, Tax, David M. J.: Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB. Wiley. ▪ Bishop, Christopher M.: Neural Networks for Pattern Recognition. Oxford University Press.		
Examination prerequisite	☐ Regular completion of homework assignments (with a presentation, if applicable)		
	☒ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☐ Passing proficiency assessments		
Examination format	☐ Oral exam		
	☐ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☒ Project (with a presentation, if applicable)		
Eligibility	Computer Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence and Data Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence in Computer Science	☒ Compulsory ☐ Elective	
Offer	☒ Summer term ☐ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Prof. Dr. Jörg Lohscheller		
Person in charge	Prof. Dr. Jörg Lohscheller		
Date of change	29.06.2026		

Maschinelles Lernen und Neuronale Netze (D)			
Inhalte	Die Methoden des maschinellen Lernens bilden eine zentrale Grundlage intelligenter Systeme und zahlreicher industrieller Anwendungen. Im Mittelpunkt stehen das Trainieren datenbasierter Modelle sowie die Analyse, Vorverarbeitung und Dimensionsreduktion von Daten. Die behandelten Verfahren werden zur Lösung von Klassifikations-, Regressions- und Clusterungsaufgaben eingesetzt; Verfahren der Dimensionsreduktion dienen zudem der Visualisierung und explorativen Analyse komplexer Datenstrukturen. Im Bereich des maschinellen Lernens werden grundlegende Konzepte behandelt wie, Datenvorverarbeitung und Dimensionsreduktion, Verfahren des unüberwachten Lernens wie k-Means, PCA, MDS, Sammon-Transformation und t-SNE sowie Verfahren des überwachten Lernens wie k-Nearest Neighbour, Bayes-Klassifikator, Entscheidungsbäume, Lineare Diskriminanzanalyse und Support Vector Machines. Ergänzend werden lineare und nichtlinear separierbare Probleme, quadratische Fehlerminimierung, Gradientenabstiegsverfahren sowie die Evaluierung von Lernverfahren betrachtet. Im Bereich neuronaler Netze werden grundlegende Architekturen, Lernprinzipien und Trainingsverfahren behandelt. Dazu gehören der Aufbau künstlicher Neuronen, Aktivierungsfunktionen, Feedforward-Netze, Backpropagation, Verlustfunktionen, Regularisierung und Optimierungsverfahren. Darüber hinaus werden zentrale Netzarchitekturen wie Fully Connected Networks, Convolutional Neural Networks bis hin zu Transformer-Modelle vorgestellt und hinsichtlich ihrer Einsatzbereiche, Trainingsanforderungen und Leistungsfähigkeit betrachtet.		
Lernergebnisse	Die Studierenden kennen grundlegende Konzepte, zentrale Verfahren und den aktuellen Forschungsstand im Bereich Maschinelles Lernen und Neuronale Netze. Sie sind fähig, aktuelle Entwicklungen zu verstehen, zu analysieren und kritisch zu reflektieren. Darüber hinaus können sie ausgewählte Verfahren eigenständig implementieren, auf geeignete Daten anwenden, deren Ergebnisse auswerten und die Leistungsfähigkeit der eingesetzten Methoden beurteilen.		
Lehrform	☒ Vorlesung		
	☐ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ Duda, Richard O.: Pattern Classification: Pattern Classification. Wiley-Interscience. ▪ Peter Flach: Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press. ▪ Ferdinand van der Heijden, de Ridder, Dick, Tax, David M. J.: Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB. Wiley. ▪ Bishop, Christopher M.: Neural Networks for Pattern Recognition. Oxford University Press.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik	☒ Pflicht ☐ Wahlpflicht	
	Künstliche Intelligenz und Data Science	☒ Pflicht ☐ Wahlpflicht	
	Artificial Intelligence in Computer Science	☒ Pflicht ☐ Wahlpflicht	
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Jörg Lohscheller		
Modulverantwortliche(r)	Prof. Dr. Jörg Lohscheller		
Änderungsdatum	29.06.2026		

Master's Thesis with Oral Defence (E)			
Contents	The final thesis involves tackling a challenging research topic, the level of difficulty of which corresponds to the future professional practice of a Master's graduate, using academic methods. The topic may cover theoretical, experimental, empirical or practice-oriented problems. In the process, students practise systematic approaches and effective working techniques, whilst establishing links to areas of application within computer science. Students present their findings in a written report, adhering to the criteria of academic work, and in a colloquium before an examination board. During this, the content of the final thesis is scrutinised within the context of the respective topic.		
Learning outcomes	By successfully completing this assignment, students have demonstrated that they are capable of independently addressing a major specialist problem using academic methods within a specified timeframe. They possess broad and integrated knowledge, including the academic fundamentals, as well as a critical understanding of the most important theories and methods. They are able to apply the knowledge, skills, methods and tools acquired during their studies to new problems and, furthermore, to independently expand upon, evaluate and interpret relevant content from an academic perspective. On this basis, they derive well-founded approaches and formulate a solution to the academic problem that is in line with the current state of the art. Furthermore, they can document their results professionally, applying the criteria of scientific practice, present them at a colloquium and defend them with sound reasoning.		
Learning format	☐ Lecture		
	☐ Exercise		
	☐ Seminar/Seminar-style class		
	☐ Lab		
	☒ Project		
Recommended prerequisites	Competencies in accordance with the learning outcomes of the modules "Seminar" and "Project". Additional recommended prerequisites depend on the assignment and are determined by the advisor.		
Reading list	▪ Booth, Wayne C.; Colomb, Gregory G.; Williams, Joseph M.; Bizup, Joseph; Fitzgerald, William T. (2016): The Craft of Research. 4th edition. University of Chicago Press. ▪ Zobel, Justin (2014): Writing for Computer Science. 3rd edition. Springer. ▪ Turabian, Kate L.; Booth, Wayne C.; Colomb, Gregory G.; Williams, Joseph M.; Bizup, Joseph (2018): A Manual for Writers of Research Papers, Theses, and Dissertations. 9th edition. University of Chicago Press. ▪ Evans, David; Gruba, Paul; Zobel, Justin (2014): How to Write a Better Thesis. 3rd edition. Springer.		
Examination prerequisite	☐ Regular completion of homework assignments (with a presentation, if applicable)		
	☐ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☐ Passing proficiency assessments		
Examination format	☐ Oral exam		
	☐ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☒ Project (with a presentation, if applicable)		
Eligibility	Computer Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence and Data Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence in Computer Science	☒ Compulsory ☐ Elective	
Offer	☒ Summer term ☒ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	25	30 hours	720 hours
Lecturer	Lecturers in the Department of Computer Science		
Person in charge	Head of study field Computer Science		
Date of change	18.06.2026		

Mixed Reality (D)	
Inhalte	▪ Grundlagen und Begrifflichkeit: das Realitäts-Virtualitäts-Kontinuum sowie die Definitionen von Augmented, Virtual, Mixed und Extended Reality (AR/VR/MR/XR); die Konzepte Immersion, Präsenz und Embodiment ▪ Display- und Gerätetechnologien: optische und videobasierte See-Through-Head-Mounted-Displays, Passthrough- und Spatial-Computing-Geräte, projektionsbasiertes und handgehaltenes AR ▪ Tracking, Registrierung und Szenenverständnis: Inside-Out-Tracking und SLAM, Sensorfusion, räumliche Anker (Spatial Anchors), Tiefenerfassung, semantische Szenenrekonstruktion, Eye- und Hand-Tracking ▪ Räumliche Interaktion: 3D-Interaktionstechniken, Gesten-, Blick-, Sprach- und tangible Eingabe, Haptik und multimodale Interaktion ▪ Echtzeit-Rendering für MR: Verdeckungsbehandlung (Occlusion), Schätzung der Beleuchtung, visuelle Kohärenz sowie Performance- und Latenzanforderungen ▪ Content-Erstellung und Plattformen: Echtzeit-Engines (z. B. Unity, Unreal), der OpenXR-Standard und gängige MR-Toolkits, Authoring-Pipelines ▪ Aktuelle Forschung und Trends: KI-gestützte MR (Neural Rendering, generative 3D-Inhalte, Foundation Models für das Szenenverständnis), Avatare und soziale/kollaborative MR, Remote-Assistance, Digitale Zwillinge und Spatial Computing ▪ Human Factors und gesundheitspsychologische Mechanismen: Präsenz, Immersion und Embodiment als psychologische Prozesse; Cybersickness, visuelle Ermüdung, kognitive Belastung, Ergonomie und Barrierefreiheit ▪ MR für Gesundheit und Wohlbefinden: Expositionstherapie, Angst- und Stressreduktion, Schmerzablenkung und -management, neurologische und motorische Rehabilitation, Exergames und Verhaltensänderung; zugrunde liegende Theorien (z. B. Selbstbestimmungstheorie, Flow, verkörperte Kognition) ▪ Verantwortungsvolle MR: psychologische Risiken und Nutzersicherheit, Schutz biometrischer und verhaltensbezogener Daten, Ethik sowie evidenzbasierte Wirksamkeitsbewertung ▪ Evaluation und Nutzerstudien: Studiendesign, validierte Instrumente (z. B. Präsenz- und Simulator-Sickness-Fragebögen, Usability- und User-Experience-Maße) sowie die Analyse der Ergebnisse
Lernergebnisse	Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, ▪ die grundlegenden Konzepte der Mixed Reality – darunter das Realitäts-Virtualitäts-Kontinuum, Präsenz, Immersion und Embodiment – zu erläutern und die zentralen Basistechnologien für Tracking, Registrierung, Rendering und räumliche Interaktion zu beschreiben; ▪ interaktive Mixed-Reality-Prototypen mit aktuellen Echtzeit-Engines und Standard-Frameworks (z. B. OpenXR) umzusetzen und geeignete Interaktions- und Visualisierungstechniken auf eine gegebene Problemstellung anzuwenden; ▪ Mixed-Reality-Systeme und Nutzererfahrungen hinsichtlich perzeptueller, ergonomischer, kognitiver und gesundheitsbezogener Faktoren zu analysieren und konkurrierende technologische und methodische Ansätze zu unterscheiden; ▪ die Eignung, Wirksamkeit sowie die ethischen und gesundheitspsychologischen Implikationen von Mixed-Reality-Anwendungen – insbesondere in Gesundheits- und Wohlbefindenskontexten – auf Basis aktueller empirischer Evidenz und validierter Evaluationsinstrumente kritisch zu bewerten; ▪ ein eigenständiges Mixed-Reality-Konzept oder einen Prototyp zu entwerfen und zu entwickeln, das technische, perzeptuelle und gesundheitspsychologische Aspekte integriert sowie ein geeignetes Evaluationskonzept umfasst, und die Entwurfsentscheidungen vor dem Hintergrund des aktuellen Forschungsstands zu begründen.
Lehrform	☒ Vorlesung
	☐ Übung
	☐ Seminar/Seminaristischer Unterricht
	☒ Labor
	☒ Projekt
Empfohlene Voraussetzungen	Grundkenntnisse in den Bereichen 3D-Computergrafik und Mensch-Computer-Interaktion sowie in der objektorientierten Programmierung; Vorkenntnisse im Umgang mit einer Echtzeit-Engine (z. B. Unity oder Unreal) sind von Vorteil, aber keine Voraussetzung.
Literatur	▪ Schmalstieg, D., & Höllerer, T. (2016). Augmented Reality: Principles and Practice. Addison-Wesley Professional ▪ LaViola, J. J., Jr., Kruijff, E., McMahan, R. P., Bowman, D. A., & Poupyrev, I. (2017). 3D User Interfaces: Theory and Practice (2nd ed.). Addison-Wesley Professional ▪ Jerald, J. (2015). The VR Book: Human-Centered Design for Virtual Reality. Association for Computing Machinery and Morgan & Claypool Publishers ▪ Milgram, P., & Kishino, F. (1994). A taxonomy of mixed reality visual displays. IEICE Transactions on Information and Systems, E77-D(12), 1321–1329 ▪ Billinghurst, M., Clark, A., & Lee, G. (2015). A survey of augmented reality. Foundations and Trends in Human–Computer Interaction, 8(2–3), 73–272. https://doi.org/10.1561/1100000049 ▪ Slater, M., & Sanchez-Vives, M. V. (2016). Enhancing our lives with immersive virtual reality. Frontiers in Robotics and AI, 3, Article 74. https://doi.org/10.3389/frobt.2016.00074 ▪ Kilteni, K., Groten, R., & Slater, M. (2012). The sense of embodiment in virtual reality. Presence: Teleoperators and Virtual Environments, 21(4), 373–387. https://doi.org/10.1162/PRES_a_00124

	▪ Freeman, D., Reeve, S., Robinson, A., Ehlers, A., Clark, D., Spanlang, B., & Slater, M. (2017). Virtual reality in the assessment, understanding, and treatment of mental health disorders. Psychological Medicine, 47(14), 2393–2400. https://doi.org/10.1017/S003329171700040X ▪ Mallari, B., Spaeth, E. K., Goh, H., & Boyd, B. S. (2019). Virtual reality as an analgesic for acute and chronic pain in adults: A systematic review and meta-analysis. Journal of Pain Research, 12, 2053–2085. https://doi.org/10.2147/JPR.S200498 ▪ Riva, G., Wiederhold, B. K., & Mantovani, F. (2019). Neuroscience of virtual reality: From virtual exposure to embodied medicine. Cyberpsychology, Behavior, and Social Networking, 22(1), 82–96. https://doi.org/10.1089/cyber.2017.29099.gri ▪ Kennedy, R. S., Lane, N. E., Berbaum, K. S., & Lilienthal, M. G. (1993). Simulator Sickness Questionnaire: An enhanced method for quantifying simulator sickness. The International Journal of Aviation Psychology, 3(3), 203–220. https://doi.org/10.1207/s15327108ijap0303_3		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik		☒ Pflicht ☐ Wahlpflicht
	Künstliche Intelligenz und Data Science		☐ Pflicht ☒ Wahlpflicht
	Artificial Intelligence in Computer Science		☐ Pflicht ☒ Wahlpflicht
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	N.N.		
Modulverantwortliche(r)	N.N.		
Änderungsdatum	30.06.2026		

Mobile Robotics (E)				
Contents	▪ Coordinate transformations for localisation ▪ Inertial navigation ▪ Calibration of accelerometers and angular rate sensors ▪ Technology and calibration ▪ Algorithms for strap-down methods ▪ Satellite navigation (signals and systems) ▪ Extensions (e.g. RTK) ▪ Data fusion of INS/GNSS (loosely- and tightly-coupled architectures) in theory (error state) and practice (simulation environment) ▪ Visual navigation using the example of visual odometry (VO) ▪ Possibilities for integrating inertial VO and INS/GNSS/VO			
Learning outcomes	Students will be able to ▪ apply geodetic transformations correctly, ▪ formulate error models for the calibration of inertial navigation systems, ▪ implement strap-down methods, ▪ explain satellite navigation at component level, ▪ apply satellite-based localisation in theory and practice, ▪ implement data fusion architectures (loosely- and tightly-coupled), ▪ understand and implement the fundamentals of visual odometry, and ▪ design and discuss INS/GNSS/VO architectures.			
Learning format	☒ Lecture			
	☒ Exercise			
	☐ Seminar/Seminar-style class			
	☐ Lab			
	☐ Project			
Recommended prerequisites	Competencies in accordance with the learning outcomes of the Bachelor's modules "Introduction to Robotics", " Robot Vision" and "Cognitive Vision Systems"			
Reading list	▪ R. Siegwart et al: Autonomous Mobile Robotics. Springer, 2. Auflage (2011). ▪ J. Hertzberg et al: Mobile Robotik. Springer, 2012. ▪ Groves: Navigation Systems. Artech House, 2013. ▪ Wendel: Integrierte Navigationssysteme. Oldenbourg, 2011. ▪ Jekeli: Inertial Navigation Systems with Geodetic Applications. De Gruyter, 2000. ▪ R. M. Rogers: Applied Mathematics in INS. Transatlantic Publishers, 2007. ▪ Titterton and Weston: Strapdown Inertial Navigation Technology. IET, 2. Auflage (2005). ▪ J. Graf: Vorlesungsunterlagen zum Modul Visuelle Navigation			
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)			
	☒ Completion of term paper/lab assignments (with a presentation, if applicable)			
	☒ Passing proficiency assessments			
Examination format	☒ Oral exam (if there are few participants)			
	☒ Written exam			
	☐ Computer-based exam			
	☐ Term paper (with a presentation, if applicable)			
	☐ Project (with a presentation, if applicable)			
Eligibility	Computer Science		☐ Compulsory ☒ Elective	
	Artificial Intelligence and Data Science		☐ Compulsory ☒ Elective	
	Artificial Intelligence in Computer Science		☐ Compulsory ☒ Elective	
Offer	☒ Summer term ☐ Winter term ☐ Irregular			
Workload	ECTS credits	Contact time	Self-study	
	5	60 hours	90 hours	
Lecturer	Prof. Dr. Jürgen Graf			
Person in charge	Prof. Dr. Jürgen Graf			
Date of change	07.07.2026			

Optimierung (D)		

Inhalte	▪ Einführung in die mathematische Optimierung
	  ▪ Optimierungsprobleme und Anwendungsgebiete
	  ▪ Klassifikation von Optimierungsproblemen
	  ▪ Konvexe und diskrete Optimierung
	▪ Lineare Programmierung (LP)
	  ▪ Formulierung linearer Optimierungsmodelle
	  ▪ Standardform und geometrische Interpretation
	  ▪ Dualitätstheorie
	  ▪ Sensitivitäts- und Postoptimalitätsanalyse
	▪ Ganzzahlige und gemischt-ganzzahlige Optimierung (IP/MIP)
	  ▪ Modellierung mit binären und ganzzahligen Variablen
	  ▪ Typische Modellierungsmuster (Auswahl, Zuordnung, Scheduling, Netzwerkprobleme)
	  ▪ Relaxationen und Schranken
	▪ Modellierung komplexer Optimierungsprobleme
	  ▪ Formulierung realer Anwendungsprobleme als LP/MIP
	  ▪ Linearisierung nichtlinearer Zusammenhänge
	  ▪ Big-M-Techniken, Indikatorvariablen
	  ▪ Mehrzieloptimierung (Überblick)
	▪ Lösungsverfahren
	  ▪ Simplex-Verfahren
	  ▪ Innere-Punkte-Verfahren (Überblick)
	  ▪ Branch-and-Bound
	  ▪ Branch-and-Cut
	  ▪ Schnittebenenverfahren (Cutting Planes)
	  ▪ Heuristiken und Metaheuristiken (Überblick)
	▪ Optimierungssolver und Modellierungssprachen
	  ▪ Einsatz moderner Solver (z. B. Gurobi)
	  ▪ Interpretation von Solver-Ausgaben
	  ▪ Parametrisierung und Performanceanalyse
	▪ Anwendungen, z.B.
	  ▪ Produktionsplanung
	  ▪ Logistik und Routing
	  ▪ Ressourcenallokation
	  ▪ Stunden- und Einsatzplanung
	  ▪ Netzwerkoptimierung	

Lernergebnisse	Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, ▪ Optimierungsprobleme aus praktischen Anwendungen zu analysieren und mathematisch zu modellieren, ▪ lineare und gemischt-ganzzahlige Optimierungsmodelle selbstständig zu formulieren, ▪ geeignete Lösungsverfahren auszuwählen und deren Eigenschaften zu beurteilen, ▪ professionelle Optimierungssolver zur Lösung komplexer Probleme einzusetzen, ▪ Solver-Ergebnisse hinsichtlich Optimalität, Laufzeit und Modellqualität kritisch zu interpretieren und ▪ Optimierungsmodelle für neue Anwendungsfälle zu entwickeln und zu evaluieren.	
Lehrform	☒ Vorlesung	
	☒ Übung	
	☐ Seminar/Seminaristischer Unterricht	
	☐ Labor	
	☐ Projekt	
Empfohlene Voraussetzungen	Grundkenntnisse der Python-Programmierung, Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „Lineare Algebra"	
Literatur	▪ Vasek Chvatal: Linear programming. A Series of Books in the Mathematical Sciences, 1983. ▪ Dimitris Bertsimas and John N. Tsitsiklis: Introduction to Linear Optimization. January 1997.	
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)	
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)	
	☐ Bestehen von Leistungsstandkontrollen	
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)	
	☒ Klausur	
	☐ Prüfung am PC	
	☐ Hausarbeit (ggf. mit Präsentation)	
	☐ Projekt (ggf. mit Präsentation)	
Verwendbarkeit	Informatik	☐ Pflicht ☒ Wahlpflicht

	Künstliche Intelligenz und Data Science		☒ Pflicht ☐ Wahlpflicht
	Artificial Intelligence in Computer Science		☐ Pflicht ☒ Wahlpflicht
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Heinz Schmitz		
Modulverantwortliche(r)	Prof. Dr. Heinz Schmitz		
Änderungsdatum	22.06.2026		

Professional Communication in German (E)			
Contents	▪ Advanced German communication for professional, technical and workplace contexts, leading towards B1+/B2. ▪ Professional language functions: presenting competences, describing responsibilities, explaining technical decisions, reporting progress, negotiating tasks, giving and receiving feedback. ▪ Workplace communication: meetings, minutes, telephone and video calls, scheduling, project coordination, conflict-sensitive communication and intercultural aspects of German workplaces. ▪ Career-oriented writing: CV elements, motivation letters, professional e-mails, internship/job application communication, LinkedIn/Xing-style profiles and short reports. ▪ Technical communication in computer science: explaining software functions, requirements, user stories, bug reports, documentation excerpts, data protection and responsible AI topics in accessible German. ▪ B1+/B2 language consolidation: complex sentence structures, passive and subjunctive forms in polite requests, connectors for argumentation, register, style and accuracy.		
Learning outcomes	Upon completion of the course, students will be able to ▪ communicate effectively in German in common professional and technical situations related to computer science and applied AI. ▪ understand the main content of workplace-related conversations, meetings, presentations and written communication when topics are familiar or prepared. ▪ write clear professional e-mails, short reports, application-related documents and technical explanations using appropriate register. ▪ present a technical or project-related topic in German in a structured way and respond to questions and feedback. ▪ participate in meetings and project discussions by explaining ideas, asking for clarification, agreeing or disagreeing politely and summarising decisions. ▪ demonstrate language competence approaching B1+/B2 of the CEFR and develop strategies for further independent progress towards B2.		
Learning format	☐ Lecture		
	☒ Exercise		
	☒ Seminar/Seminar-style class		
	☐ Lab		
	☒ Project		
Recommended prerequisites	Competencies in accordance with the learning outcomes of the module "Academic Communication in German"		
Reading list	▪ Aspekte neu B1 plus / B2 or equivalent CEFR-oriented material for professional communication ▪ Mit Erfolg zu telc Deutsch B2 / Goethe-Zertifikat B2 or comparable practice material ▪ Selected authentic professional materials: job advertisements, CV and cover letter examples, meeting agendas, reports and technical documentation ▪ University-provided materials on applications, internships, workplace culture and communication in German companies		
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)		
	☒ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☒ Passing proficiency assessments		
Examination format	☒ Oral exam		
	☐ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☐ Project (with a presentation, if applicable)		
Eligibility	Computer Science		☐ Compulsory ☐ Elective
	Artificial Intelligence and Data Science		☐ Compulsory ☐ Elective
	Artificial Intelligence in Computer Science		☒ Compulsory ☐ Elective (non-native speakers only)
Offer	☐ Summer term ☒ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Contract lecturer		
Person in charge	Prof. Dr. Georg Schneider		

Date of change	08.07.2026

<table>
<tr><td colspan="5">Project (E)</td></tr>
<tr><td>Contents</td><td colspan="4">The Project involves tackling a challenging, real-world problem under the close supervision of a lecturer. The level of difficulty corresponds to the professional practice that a Master's graduate will encounter later on. The project focuses on developing systematic approaches and effective working techniques, whilst establishing links to areas of application within computer science. Students present their findings in a written report, applying the criteria of academic writing.</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">By successfully completing this assignment, students have demonstrated that they are able, within a specified timeframe, to familiarise themselves independently with an area of application within computer science and to tackle a practical problem. They possess broad and integrated knowledge, as well as a critical understanding of the most important theories and methods. They are able to apply the knowledge, skills, methods and tools acquired during their studies to specific problems. On this basis, they derive well-founded approaches and formulate a state-of-the-art solution to the practical problem. They understand the factors contributing to success and are proficient in methods for defining, planning and monitoring projects. Consequently, they can structure complex tasks effectively and manage typical interface problems at both a technical/specialist and a social level. They can document their work professionally, applying the criteria of academic practice.</td></tr>
<tr><td rowspan="5">Learning format</td><td colspan="4">☐ Lecture</td></tr>
<tr><td colspan="4">☐ Exercise</td></tr>
<tr><td colspan="4">☐ Seminar/Seminar-style class</td></tr>
<tr><td colspan="4">☐ Lab</td></tr>
<tr><td colspan="4">☒ Project</td></tr>
<tr><td>Recommended prerequisites</td><td colspan="4">Depend on the assignment and are determined by the advisor.</td></tr>
<tr><td>Reading list</td><td colspan="4">▪ Booth, Wayne C.; Colomb, Gregory G.; Williams, Joseph M.; Bizup, Joseph; Fitzgerald, William T. (2016): The Craft of Research. 4th edition. University of Chicago Press.
▪ Zobel, Justin (2014): Writing for Computer Science. 3rd edition. Springer.
▪ Turabian, Kate L.; Booth, Wayne C.; Colomb, Gregory G.; Williams, Joseph M.; Bizup, Joseph (2018): A Manual for Writers of Research Papers, Theses, and Dissertations. 9th edition. University of Chicago Press.
▪ Evans, David; Gruba, Paul; Zobel, Justin (2014): How to Write a Better Thesis. 3rd edition. Springer.</td></tr>
<tr><td rowspan="3">Examination prerequisite</td><td colspan="4">☐ Regular completion of homework assignments (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☐ Completion of term paper/lab assignments (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☐ Passing proficiency assessments</td></tr>
<tr><td rowspan="5">Examination format</td><td colspan="4">☐ Oral exam</td></tr>
<tr><td colspan="4">☐ Written exam</td></tr>
<tr><td colspan="4">☐ Computer-based exam</td></tr>
<tr><td colspan="4">☐ Term paper (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☒ Project (with a presentation, if applicable)</td></tr>
<tr><td rowspan="3">Eligibility</td><td colspan="3">Computer Science</td><td>☒ Compulsory ☐ Elective</td></tr>
<tr><td colspan="3">Artificial Intelligence and Data Science</td><td>☒ Compulsory ☐ Elective</td></tr>
<tr><td colspan="3">Artificial Intelligence in Computer Science</td><td>☒ Compulsory ☐ Elective</td></tr>
<tr><td>Offer</td><td colspan="4">☒ Summer term ☒ Winter term ☐ Irregular</td></tr>
<tr><td rowspan="2">Workload</td><td>ECTS credits</td><td>Contact time</td><td colspan="2">Self-study</td></tr>
<tr><td>15</td><td>30 hours</td><td colspan="2">270 hours</td></tr>
<tr><td>Lecturer</td><td colspan="4">Lecturers in the Department of Computer Science</td></tr>
<tr><td>Person in charge</td><td colspan="4">Head of study field Computer Science</td></tr>
<tr><td>Date of change</td><td colspan="4">18.06.2026</td></tr>
</table>

Projektarbeit (D)			
Inhalte	Die Projektarbeit umfasst die Bearbeitung einer qualifizierten Aufgabenstellung aus der Praxis unter intensiver Betreuung durch einen Dozenten. Der Schwierigkeitsgrad entspricht der späteren Berufspraxis eines Master-Absolventen. Dabei werden systematische Vorgehensweisen und sinnvolle Arbeitstechniken sowie die Verbindung zu Anwendungsgebieten der Informatik hergestellt. Die Studierenden präsentieren ihre Ergebnisse in einer Ausarbeitung unter Anwendung der Kriterien wissenschaftlichen Arbeitens.		
Lernergebnisse	Die Studierenden haben durch die erfolgreiche Bearbeitung gezeigt, dass sie in der Lage sind, innerhalb einer vorgegebenen Frist sich selbständig in ein Anwendungsgebiet der Informatik einzuarbeiten und ein praktisches Problem zu bearbeiten. Sie verfügen über ein breites und integriertes Wissen sowie über ein kritisches Verständnis der wichtigsten Theorien und Methoden. Sie sind in der Lage, die im Studium erworbenen Kenntnisse, Fähigkeiten, Methoden und Werkzeuge auf vorgegebene Fragestellungen anzuwenden. Sie leiten auf dieser Basis fundierte Lösungsansätze ab und formulieren eine dem Stand der Technik entsprechende Lösung für das praktische Problem. Sie verstehen die Erfolgsfaktoren und beherrschen Methoden der Definition, Planung sowie Kontrolle von Projekten. Daher können sie komplexe Aufgaben sinnvoll strukturieren und typische Schnittstellenprobleme sowohl auf technisch-fachlicher als auch auf sozialer Ebene bewältigen. Sie können Arbeitsergebnisse professionell und unter Anwendung der Kriterien des wissenschaftlichen Arbeitens dokumentieren.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☒ Projekt		
Empfohlene Voraussetzungen	Abhängig von der Aufgabenstellung; werden vom Betreuer festgelegt.		
Literatur	▪ Helmut Balzert, Marion Schröder, Christian Schäfer: Wissenschaftliches Arbeiten. W3L GmbH. ▪ Booth, Wayne C.; Colomb, Gregory G.; Williams, Joseph M.; Bizup, Joseph; Fitzgerald, William T. (2016): The Craft of Research. 4. Auflage. University of Chicago Press. ▪ Zobel, Justin (2014): Writing for Computer Science. 3. Auflage. Springer. ▪ Turabian, Kate L.; Booth, Wayne C.; Colomb, Gregory G.; Williams, Joseph M.; Bizup, Joseph (2018): A Manual for Writers of Research Papers, Theses, and Dissertations. 9. Auflage. University of Chicago Press. ▪ Evans, David; Gruba, Paul; Zobel, Justin (2014): How to Write a Better Thesis. 3. Auflage. Springer.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik	☒ Pflicht ☐ Wahlpflicht	
	Künstliche Intelligenz und Data Science	☒ Pflicht ☐ Wahlpflicht	
	Artificial Intelligence in Computer Science	☒ Pflicht ☐ Wahlpflicht	
Angebot	☒ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	15	30 Stunden	270 Stunden
Lehrende(r)	Dozenten des Fachbereichs Informatik		
Modulverantwortliche(r)	Fachrichtungsleiter Informatik		
Änderungsdatum	18.06.2026		

Safety KI-basierter Cyber Physical Systems (D)			
Inhalte	▪ Definition Cyber Physical Systems (CPS) ▪ Terminologie zu Safety, Dependability, Fault-Tolerance, Real-Time ▪ Beispiele KI-basierter Cyber Physical Systems, insbesondere aus den Bereichen Advanced Driver Assistance Systems (ADAS) und Automated Driving Systems (ADS) ▪ Zusammenhang Safety und Security ▪ Herausforderung Safety und KI ▪ Gesellschaftliche Bedeutung und ethische Fragestellungen ▪ Safety of the Intended Function (SOTIF) vs Functional Safety (FuSa) ▪ Einschlägige Standards und Normen (z.B. SAE J 3016, ISO 26262, ISO/DIS 21448) ▪ Übungsprojekt am Forschungsgroßgerät dynamischer Fahrsimulator FaSiMo: KI für ADAS bzw. ADS entwickeln gemäß SOTIF		
Lernergebnisse	Die Studierenden können die besonderen Herausforderungen bei der Entwicklung sicherheitsrelevanter KI-basierter Cyber Physical Systems erläutern. Sie können erklären, welche Rolle unterschiedliche Sicherheitsnormen bei der Entwicklung KI-basierter Lösungen für das automatisierte Fahren spielen und was der Unterschied zwischen SOTIF und FuSa ist. Sie kennen die grundlegenden Schritte, die bei der Entwicklung von Deep Learning/KI-Lösungen erforderlich sind, wissen um die Limitierungen solcher Systeme hinsichtlich Safety und kennen die standardisierten Herangehensweisen gemäß SOTIF. Sie sind in der Lage in interdisziplinären Teams unter Verwendung der einschlägigen Fachbegriffe Lösungsansätze für sicherheitsrelevante KI-basierte CPS gemäß dem SOTIF Prozess zu diskutieren, analysieren, entwerfen und zu gestalten.		
Lehrform	☐ Vorlesung		
	☒ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Gute Kenntnisse der englischen Sprache; Programmierkenntnisse (z.B. in Python oder C/C++)		
Literatur	▪ ISO 21448 (wird in der Veranstaltung bereitgestellt) ▪ Algirdas Avižienis, Jean-Claude Laprie, Brian Randell and Carl Landwehr: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, 2004 & 2013.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik	☐ Pflicht ☒ Wahlpflicht	
	Künstliche Intelligenz und Data Science	☐ Pflicht ☒ Wahlpflicht	
	Artificial Intelligence in Computer Science	☐ Pflicht ☒ Wahlpflicht	
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Jörn Schneider		
Modulverantwortliche(r)	Prof. Dr. Jörn Schneider		
Änderungsdatum	18.06.2026		

Seminar (D)			
Inhalte	Das Seminar umfasst die selbstständige Erarbeitung eines vorgegebenen begrenzten Themenbereiches anhand von wissenschaftlicher Fachliteratur und anderen Quellen sowie dessen schriftliche und mündliche Darstellung. Es werden wechselnde aktuelle Themen aus der Informatik angeboten, die im Schwierigkeitsgrad für das Master-Studium angemessen sind.		
Lernergebnisse	Die Studierenden verfügen über ein vertieftes Verständnis spezifischer Theorien sowie Ansätze im gewählten Themenbereich und kennen den jeweils aktuellen Stand der Technik. Sie haben eigenständige Lernstrategien erfolgreich angewandt sowie ihre Kommunikations- und Kooperationsfähigkeit in Teams verbessert. Sie kennen wesentliche Merkmale von Präsentationstechniken und haben diese bei der verständlichen Darstellung ihrer Ergebnisse angewendet.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Abhängig vom Thema des Seminars; werden vom Betreuer festgelegt.		
Literatur	▪ Helmut Balzert, Marion Schröder, Christian Schäfer: Wissenschaftliches Arbeiten. W3L GmbH. ▪ Booth, Wayne C.; Colomb, Gregory G.; Williams, Joseph M.; Bizup, Joseph; Fitzgerald, William T. (2016): The Craft of Research. 4. Auflage. University of Chicago Press. ▪ Zobel, Justin (2014): Writing for Computer Science. 3. Auflage. Springer. ▪ Turabian, Kate L.; Booth, Wayne C.; Colomb, Gregory G.; Williams, Joseph M.; Bizup, Joseph (2018): A Manual for Writers of Research Papers, Theses, and Dissertations. 9. Auflage. University of Chicago Press. ▪ Evans, David; Gruba, Paul; Zobel, Justin (2014): How to Write a Better Thesis. 3. Auflage. Springer.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☒ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik		☒ Pflicht ☐ Wahlpflicht
	Künstliche Intelligenz und Data Science		☒ Pflicht ☐ Wahlpflicht
	Artificial Intelligence in Computer Science		☒ Pflicht ☐ Wahlpflicht
Angebot	☒ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	30 Stunden	120 Stunden
Lehrende(r)	Dozenten des Fachbereichs Informatik		
Modulverantwortliche(r)	Fachrichtungsleiter Informatik		
Änderungsdatum	18.06.2026		

Seminar (E)			
Contents	The seminar involves independent research into a specified, limited subject area, drawing on academic literature and other sources, followed by a written and oral presentation of the findings. A variety of current topics from the field of computer science are offered, all of which are of a level of difficulty appropriate for a Master's programme.		
Learning outcomes	Students have gained an in-depth understanding of specific theories and approaches within their chosen subject area and are familiar with the current state of the art. They have successfully applied independent learning strategies and improved their ability to communicate and collaborate within teams. They are familiar with the key features of presentation techniques and have applied these to present their findings clearly.		
Learning format	☐ Lecture		
	☐ Exercise		
	☒ Seminar/Seminar-style class		
	☐ Lab		
	☐ Project		
Recommended prerequisites	Depending on the topic of the seminar and are determined by the advisor.		
Reading list	▪ Booth, Wayne C.; Colomb, Gregory G.; Williams, Joseph M.; Bizup, Joseph; Fitzgerald, William T. (2016): The Craft of Research. 4th edition. University of Chicago Press. ▪ Zobel, Justin (2014): Writing for Computer Science. 3rd edition. Springer. ▪ Turabian, Kate L.; Booth, Wayne C.; Colomb, Gregory G.; Williams, Joseph M.; Bizup, Joseph (2018): A Manual for Writers of Research Papers, Theses, and Dissertations. 9th edition. University of Chicago Press. ▪ Evans, David; Gruba, Paul; Zobel, Justin (2014): How to Write a Better Thesis. 3rd edition. Springer.		
Examination prerequisite	☐ Regular completion of homework assignments (with a presentation, if applicable)		
	☐ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☐ Passing proficiency assessments		
Examination format	☐ Oral exam		
	☐ Written exam		
	☐ Computer-based exam		
	☒ Term paper (with a presentation, if applicable)		
	☐ Project (with a presentation, if applicable)		
Eligibility	Computer Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence and Data Science	☒ Compulsory ☐ Elective	
	Artificial Intelligence in Computer Science	☒ Compulsory ☐ Elective	
Offer	☐ Summer term ☒ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	30 hours	120 hours
Lecturer	Lecturers in the Department of Computer Science		
Person in charge	Head of study field Computer Science		
Date of change	18.06.2026		

<table>
<tr><td colspan="5">Simulation Technology and Reinforcement Learning (E)</td></tr>
<tr><td>Contents</td><td colspan="4">▪ Modelling and Simulation
▪ Project workflow
▪ Discrete simulation
▪ Continuous simulation
▪ Empirical models
▪ Reinforcement Learning
▪ Fundamentals
▪ Tabular methods
▪ Approximation using neural networks
▪ Actor Critic
▪ Alpha Zero</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">Students will be able to
▪ explain the basic concepts of modelling and simulation, as well as reinforcement learning,
▪ program the techniques presented, and
▪ apply the techniques presented in a practical project.</td></tr>
<tr><td rowspan="5">Learning format</td><td colspan="4">☒ Lecture</td></tr>
<tr><td colspan="4">☐ Exercise</td></tr>
<tr><td colspan="4">☐ Seminar/Seminar-style class</td></tr>
<tr><td colspan="4">☐ Lab</td></tr>
<tr><td colspan="4">☒ Project</td></tr>
<tr><td>Recommended prerequisites</td><td colspan="4">▪ Knowledge of Python
▪ Knowledge of applied mathematics
▪ Basic knowledge of neural networks</td></tr>
<tr><td>Reading list</td><td colspan="4">▪ Louis G. Birta, Gilbert Arbez: Modelling and Simulation, Exploring Dynamic Systems Behaviour, Springer 2019
▪ Richard S. Sutton, Andrew G.Barto: Reinforcement Learning, MIT Press, 2020
▪ Maxim Lapan: Deep Reinforcement Learning Hands-On, Packt 2018</td></tr>
<tr><td rowspan="3">Examination prerequisite</td><td colspan="4">☐ Regular completion of homework assignments (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☒ Completion of term paper/lab assignments (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☐ Passing proficiency assessments</td></tr>
<tr><td rowspan="5">Examination format</td><td colspan="4">☐ Oral exam</td></tr>
<tr><td colspan="4">☐ Written exam</td></tr>
<tr><td colspan="4">☐ Computer-based exam</td></tr>
<tr><td colspan="4">☐ Term paper (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☒ Project (with a presentation, if applicable)</td></tr>
<tr><td rowspan="3">Eligibility</td><td colspan="3">Computer Science</td><td>☐ Compulsory ☒ Elective</td></tr>
<tr><td colspan="3">Artificial Intelligence and Data Science</td><td>☒ Compulsory ☐ Elective</td></tr>
<tr><td colspan="3">Artificial Intelligence in Computer Science</td><td>☒ Compulsory ☐ Elective</td></tr>
<tr><td>Offer</td><td colspan="4">☐ Summer term ☒ Winter term ☐ Irregular</td></tr>
<tr><td rowspan="2">Workload</td><td>ECTS credits</td><td>Contact time</td><td colspan="2">Self-study</td></tr>
<tr><td>5</td><td>60 hours</td><td colspan="2">90 hours</td></tr>
<tr><td>Lecturer</td><td colspan="4">Prof. Dr. Christoph Lürig</td></tr>
<tr><td>Person in charge</td><td colspan="4">Prof. Dr. Christoph Lürig</td></tr>
<tr><td>Date of change</td><td colspan="4">18.06.2026</td></tr>
</table>

Simulationstechnik und Reinforcement Learning (D)			

Inhalte	▪ Modelling und Simulation
	  ▪ Projektablauf
	  ▪ Diskrete Simulation
	  ▪ Kontinuierliche Simulation
	  ▪ Empirische Modelle
	▪ Reinforcement Learning
	  ▪ Grundlagen
	  ▪ Tabellenverfahren
	  ▪ Approximation mit Neuronalen Netzen
	  ▪ Actor Critic
	  ▪ Alpha Zero		

Lernergebnisse	Die Studierenden können ▪ die Grundkonzepte von Modelling und Simulation sowie Reinforcement Learning erklären, ▪ die vorgestellten Techniken programmieren, sowie ▪ die vorgestellten Techniken in einem praktischen Projekt umsetzen.		
Lehrform	☒ Vorlesung		
	☐ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☒ Projekt		
Empfohlene Voraussetzungen	▪ Kenntnisse in Python ▪ Kenntnisse in angewandter Mathematik ▪ Grundlagenkenntnisse in Neuronalen Netzen		
Literatur	▪ Louis G. Birta, Gilbert Arbez: Modelling and Simulation, Exploring Dynamic Systems Behaviour, Springer 2019 ▪ Richard S. Sutton, Andrew G.Barto: Reinforcement Learning, MIT Press, 2020 ▪ Maxim Lapan: Deep Reinforcement Learning Hands-On, Packt 2018		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik		☐ Pflicht ☒ Wahlpflicht
	Künstliche Intelligenz und Data Science		☒ Pflicht ☐ Wahlpflicht
	Artificial Intelligence in Computer Science		☒ Pflicht ☐ Wahlpflicht
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Christoph Lürig		
Modulverantwortliche(r)	Prof. Dr. Christoph Lürig		
Änderungsdatum	18.06.2026		

Software-Architekturen (D)			
Inhalte	Das Modul befasst sich sowohl theoretisch als auch praktisch mit dem Entwurf, der Dokumentation, der Analyse und dem Management von Software-Architekturen. Der erste Teil der Vorlesung führt dazu in grundlegende Begriffe und Konzepte ein, erläutert die Rolle des Software-Architekten in der Software-Entwicklung und stellt Notationen für Software-Architekturen vor. Der zweite Teil der Vorlesung demonstriert Vorgehensweisen, Stile und Muster für die Erstellung, Bewertung und Verwaltung von Software-Architekturen. Die Übung ergänzt beide Teile durch theoretische und praktische Aufgaben. Im dritten Teil des Moduls stellen die Studierenden selbst erarbeitete Inhalte zu vertiefenden Themen in kurzen Referaten vor.		
Lernergebnisse	Die Studierenden können ▪ die Rolle des Software-Architekten in der Software-Entwicklung beschreiben, ▪ grundlegende Begriffe und Konzepte der Software-Architektur beschreiben und gegeneinander abgrenzen, ▪ Techniken zur Notation von Software-Architekturen anwenden, ▪ Verfahren des Architekturentwurfs, der Architekturbewertung und des Managements von Unternehmensarchitekturen beschreiben und anwenden, ▪ Architekturstil und -muster benennen, zueinander in Beziehung setzen und gegeneinander abgrenzen, ▪ ausgewählte Themen der Software-Architektur eigenständig erarbeiten und präsentieren, ▪ für gegebene Szenarien Entwurfsalternativen abwägen, passende Architekturentscheidungen treffen und Architekturen dokumentieren.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Bachelor-Module „Objektorientierte Programmierung - Einführung", „Software-Entwurf" und „Grundlagen des Anforderungsmanagements"		
Literatur	▪ Nick Rozanski, Eoin Woods: Software Systems Architecture. 2. Auflage, Addison-Wesley Professional, 2011. ▪ Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice. 3. Auflage, Addison-Wesley Longman, 2012. ▪ Ralf Reussner, Wilhelm Hasselbring: Handbuch der Software-Architektur. 2. Auflage, dpunkt.verlag, 2008.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik		☒ Pflicht ☐ Wahlpflicht
	Künstliche Intelligenz und Data Science		☐ Pflicht ☒ Wahlpflicht
	Artificial Intelligence in Computer Science		☐ Pflicht ☒ Wahlpflicht
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Michael Striewe		
Modulverantwortliche(r)	Prof. Dr. Michael Striewe		
Änderungsdatum	18.06.2026		

Software-Qualitätsmanagement (D)			

Inhalte	▪ Einführung und Überblick
	▪ Qualitätssicherung
	▪ Manuelle Prüfmethoden
	▪ Verbesserung der Prozessqualität
	▪ Produktqualität - Komponenten
	  ▪ Testende Verfahren
	  ▪ Verifizierende Verfahren
	  ▪ Analysierende Verfahren
	▪ Produktqualität - Systeme
	  ▪ Integrationstest
	  ▪ System- und Abnahmetest
	▪ Konfigurations- und Änderungsmanagement		

Lernergebnisse	Die Studierenden können ▪ Qualitätsbegriffe definieren und einordnen, ▪ die Prinzipien der Software-Qualitätssicherung erklären und begründen, ▪ (Code-)Inspektionen durchführen, ▪ kontrollflussorientierte und datenflussorientierte Testverfahren einsetzen, ▪ die Konzepte der Verifikation und des symbolischen Testens verwenden und gegen testende Verfahren abgrenzen, ▪ für einfache Beispiele Integrations- und Abnahmetests durchführen, ▪ Testwerkzeuge einsetzen, sowie ▪ Werkzeuge und Verfahren des Konfigurationsmanagements einsetzen.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ Roman, Adam: A Study Guide to the ISTQB® Foundation Level 2018 Syllabus: Test Techniques and Sample Mock Exams. Springer, 1. Auflage, 2018. ▪ Spillner, Andreas; Roßner, Thomas; Winter, Mario; Linz, Tilo: Praxiswissen Softwaretest – Testmanagement - Aus- und Weiterbildung zum Certified Tester – Advanced Level nach ISTQB-Standard. dpunkt Verlag, 4. Auflage, 2014. ▪ Liggesmeyer, Peter: Software-Qualität. Spektrum Verlag, 2. Auflage, 2009.		
Studienleistung	☒ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☐ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik		☒ Pflicht ☐ Wahlpflicht
	Künstliche Intelligenz und Data Science		☐ Pflicht ☒ Wahlpflicht
	Artificial Intelligence in Computer Science		☐ Pflicht ☒ Wahlpflicht
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Georg Rock		
Modulverantwortliche(r)	Prof. Dr. Georg Rock		
Änderungsdatum	18.06.2026		

Statistics and Learning Theory (D)			
Inhalte	Statistik und statistische Lerntheorie bilden die theoretische Grundlage der Bereiche Maschinelles Lernen und Data Science. In diesem Modul werden folgende Inhalte behandelt: ▪ Hypothesentests: Binomialtest, Chi-Quadrat-Test, t-Test, ANOVA, Neyman-Pearson-Lemma, Testen mehrerer Hypothesen, Entscheidungstheorie ▪ Schätzungstheorie: MLE, MAP, Fisher-Information + Cramer-Rao-Untergrenze, Anwendungen exponentieller Familien, vollständig bayesianischer Ansatz, ▪ Statistische Lerntheorie: Fehlerzerlegung und Bias-Varianz-Kompromiss, PAC-Lerngrenzen, Wachstumsfunktionen, VC-Dimension, Rademacher-Komplexität, algorithmische Stabilität, Ensemble-Methoden		
Lernergebnisse	Die Studierenden erlernen grundlegende Prinzipien sowie vertiefte Konzepte, Methoden und Techniken der Statistik und der statistischen Lerntheorie. Sie sind in der Lage, selbstständig Primärliteratur, insbesondere in englischer Sprache, zu nutzen, um fortgeschrittene wissenschaftliche Kenntnisse und Kompetenzen zu erwerben. Zudem lernen sie, die Grenzen von KI-Technologien und datenwissenschaftlichen Methoden einzuschätzen und damit verbundene Sicherheitsaspekte kritisch zu bewerten. Die Studierenden können komplexe Probleme analysieren und strukturieren, eigene Ideen entwickeln und Lösungen mithilfe geeigneter Methoden und Werkzeuge umsetzen. In diesem Modul werden die Studierenden auf weitere wissenschaftliche Qualifikationen vorbereitet.		
Lehrform	☒ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Literatur	▪ Henze, Norbert: Stochastik für Einsteiger. Springer, 13. Auflage (2021). ▪ L. Wasserman: All of Statistics. Springer, 2003. ▪ M. Hardt, B. Recht: Patterns, Predictions, and Actions: Foundations of Machine Learning. Princeton University Press, 2022.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☒ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☐ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik	☐ Pflicht ☒ Wahlpflicht	
	Künstliche Intelligenz und Data Science	☒ Pflicht ☐ Wahlpflicht	
	Artificial Intelligence in Computer Science	☒ Pflicht ☐ Wahlpflicht	
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Hans-Peter Beise		
Modulverantwortliche(r)	Prof. Dr. Hans-Peter Beise		
Änderungsdatum	18.06.2026		

<table>
<tr><td colspan="5">Statistics and Learning Theory (E)</td></tr>
<tr><td>Contents</td><td colspan="4">Statistics and statistical learning theory constitute the theoretical foundation of machine learning and data science.

The following content is covered in this module:
▪ Hypothesis tests: Binomial test, Chi-square test, t-test, ANOVA, Neyman-Pearson lemma, multi hypothesis testing, decision theory
▪ Estimation theory: MLE, MAP, Fisher information + Cramer-Rao lower bound, applications of exponential families, full-Bayesian approach,
▪ Statistical learning theory: Error decomposition and bias-variance tradeoff, PAC Learning bounds, growth functions, VC-dimension, Rademacher complexity, algorithmic stability, ensemble methods</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">Students learn fundamental principles and deeper concepts, methods and techniques in statistics and statistical learning theory. Students can independently use primary literature, especially in English, to acquire advanced scientific knowledge and competences. They also learn to evaluate the limitations of AI technologies and data science methods and to critically assess related safety and security issues. Students can analyze and structure complex problems, develop their own ideas and implement solutions by means of adequate methods and tools. In this module, students are prepared for further scientific qualifications.</td></tr>
<tr><td rowspan="5">Learning format</td><td colspan="4">☒ Lecture</td></tr>
<tr><td colspan="4">☐ Exercise</td></tr>
<tr><td colspan="4">☒ Seminar/Seminar-style class</td></tr>
<tr><td colspan="4">☐ Lab</td></tr>
<tr><td colspan="4">☐ Project</td></tr>
<tr><td>Recommended prerequisites</td><td colspan="4">None</td></tr>
<tr><td>Reading list</td><td colspan="4">▪ Henze, Norbert: Stochastik für Einsteiger. Springer, 13th edition (2021).
▪ L. Wasserman: All of Statistics. Springer, 2003.
▪ M. Hardt, B. Recht: Patterns, Predictions, and Actions: Foundations of Machine Learning. Princeton University Press, 2022.</td></tr>
<tr><td rowspan="3">Examination prerequisite</td><td colspan="4">☐ Regular completion of homework assignments (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☒ Completion of term paper/lab assignments (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☐ Passing proficiency assessments</td></tr>
<tr><td rowspan="5">Examination format</td><td colspan="4">☒ Oral exam (if there are few participants)</td></tr>
<tr><td colspan="4">☒ Written exam</td></tr>
<tr><td colspan="4">☐ Computer-based exam</td></tr>
<tr><td colspan="4">☐ Term paper (with a presentation, if applicable)</td></tr>
<tr><td colspan="4">☐ Project (with a presentation, if applicable)</td></tr>
<tr><td rowspan="3">Eligibility</td><td colspan="3">Computer Science</td><td>☐ Compulsory ☒ Elective</td></tr>
<tr><td colspan="3">Artificial Intelligence and Data Science</td><td>☒ Compulsory ☐ Elective</td></tr>
<tr><td colspan="3">Artificial Intelligence in Computer Science</td><td>☒ Compulsory ☐ Elective</td></tr>
<tr><td>Offer</td><td colspan="4">☐ Summer term ☒ Winter term ☐ Irregular</td></tr>
<tr><td rowspan="2">Workload</td><td>ECTS credits</td><td>Contact time</td><td colspan="2">Self-study</td></tr>
<tr><td>5</td><td>60 hours</td><td colspan="2">90 hours</td></tr>
<tr><td>Lecturer</td><td colspan="4">Prof. Dr. Hans-Peter Beise</td></tr>
<tr><td>Person in charge</td><td colspan="4">Prof. Dr. Hans-Peter Beise</td></tr>
<tr><td>Date of change</td><td colspan="4">18.06.2026</td></tr>
</table>

Verlässliche Echtzeitsysteme (D)			
Inhalte	▪ Dependability ▪ Terminologie (z.B. nach Laprie) ▪ Echtzeitsysteme ▪ Systemeigenschaft Sicherheit ▪ Einschlägige Normen (z.B. IEC 61508, ISO 26262, DO 178 B) ▪ Fehlervermeidung vs. Fehlertoleranz ▪ Formale Methoden ▪ Echtzeitbetriebssysteme ▪ Parallelisierung von Echtzeitsystemen ▪ Anwendungsbeispiele, z.B. ESP, Motorsteuerung, Fahrerassistenzsysteme		
Lernergebnisse	Die Studierenden haben die Prinzipien verlässlicher Echtzeitsysteme und die Besonderheiten bei deren Entwicklung kennen gelernt. Sie sind in der Lage, in interdisziplinären Teams bei der Entwicklung von sicherheitsrelevanten Echtzeitsystemen mitzuwirken.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☒ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Bachelor-Moduls „Technische Informatik“		
Literatur	▪ ISO 26262 (wird in der Veranstaltung bereitgestellt) ▪ Algirdas Avižienis, Jean-Claude Laprie, Brian Randell and Carl Landwehr: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing, 2004 & 2013.		
Studienleistung	☐ Regelmäßige Bearbeitung von Hausaufgaben (ggf. mit Präsentation)		
	☒ Bearbeitung von Haus-/Laborarbeiten (ggf. mit Präsentation)		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Mündliche Prüfung		
	☐ Klausur		
	☐ Prüfung am PC		
	☐ Hausarbeit (ggf. mit Präsentation)		
	☒ Projekt (ggf. mit Präsentation)		
Verwendbarkeit	Informatik		☐ Pflicht ☒ Wahlpflicht
	Künstliche Intelligenz und Data Science		☐ Pflicht ☒ Wahlpflicht
	Artificial Intelligence in Computer Science		☐ Pflicht ☒ Wahlpflicht
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	5	60 Stunden	90 Stunden
Lehrende(r)	Prof. Dr. Jörn Schneider		
Modulverantwortliche(r)	Prof. Dr. Jörn Schneider		
Änderungsdatum	18.06.2026		

Visual Navigation (E)			
Contents	SLAM – Front-end, Back-end, Mapping: ▪ Introduction to the subject area ▪ Representations and transformations ▪ Lie groups and Lie algebras ▪ Camera models and image formation ▪ Convex optimisation (Introduction) ▪ Visual odometry I (invariant features, epipolar geometry and registration) ▪ Visual odometry II (optical flow and direct methods) ▪ Filters and optimisation I (Bundle block adjustment and graph-based optimisation) ▪ Filters and optimisation II (Efficient solvers and pose graph optimisation) ▪ Identification of loops in trajectories ▪ (Dense) reconstruction from multiple views ▪ Visual stereo odometry and other methods		
Learning outcomes	Students will have a firm grasp of the mathematical principles underlying the design of a SLAM-based navigation system and will possess the practical skills to independently implement the basic features of a selected SLAM model (Self-Localisation and Mapping). SLAM-based navigation enables a mobile robot to autonomously build a model of its environment by exploring it with visual sensors, whilst determining the trajectory it has travelled. Students will master the analysis and development of such a system. Building on a system design presented in the lecture, various concepts and methods are discussed during the course, which are then implemented during the practical sessions using selected tasks. This ensures that students acquire the practical skills upon successful completion of the practical sessions. According to many experts, visual navigation currently represents one of the most important and fundamental challenges in robotics research and will create exciting and challenging job opportunities in the future within the field of research and development in both industrial and academic settings (e.g. in autonomous driving).		
Learning format	☒ Lecture		
	☒ Exercise		
	☐ Seminar/Seminar-style class		
	☐ Lab		
	☐ Project		
Recommended prerequisites	None		
Reading list	▪ Yi Ma et al.: An Invitation to 3-D Vision: From Images to Geometric Models. 1. Auflage (2003), Springer. ▪ Richard Hartley & Andrew Zisserman: Multiple View Geometry in computer vision. 2. Auflage (2004), Cambridge University Press. ▪ Xiang Gao & Tao Thang: Introduction to Visual SLAM: From Theory to Practice. 2021, Springer. ▪ Wolfgang Förstner & Bernhard P. Wrobel: Photogrammetric Computer Vision: Statistics, Geometry, Orientation and Reconstruction. 2018, Springer.		
Examination prerequisite	☒ Regular completion of homework assignments (with a presentation, if applicable)		
	☐ Completion of term paper/lab assignments (with a presentation, if applicable)		
	☐ Passing proficiency assessments		
Examination format	☒ Oral exam (if there are few participants)		
	☒ Written exam		
	☐ Computer-based exam		
	☐ Term paper (with a presentation, if applicable)		
	☐ Project (with a presentation, if applicable)		
Eligibility	Computer Science	☐ Compulsory ☒ Elective	
	Artificial Intelligence and Data Science	☐ Compulsory ☒ Elective	
	Artificial Intelligence in Computer Science	☐ Compulsory ☒ Elective	
Offer	☐ Summer term ☒ Winter term ☐ Irregular		
Workload	ECTS credits	Contact time	Self-study
	5	60 hours	90 hours
Lecturer	Prof. Dr. Jürgen Graf		
Person in charge	Prof. Dr. Jürgen Graf		
Date of change	06.07.2026		