TECHNISCHE UNIVERSITÄT WIEN

Bachelor

Master

Doktorat

Universitäts-lehrgang

Studienplan (Curriculum)
für das

Masterstudium

Software Engineering

UE 066 937

Technische Universität Wien
Beschluss des Senats der Technischen Universität Wien
am 16. Juni 2025

Gültig ab 1. Oktober 2025

Inhaltsverzeichnis

§ 1 Grundlage und Geltungsbereich

Der vorliegende Studienplan definiert und regelt das ingenieurwissenschaftliche, englischsprachige Masterstudium *Software Engineering* an der Technischen Universität Wien. Es basiert auf dem Universitätsgesetz 2002 BGBl. I Nr. 120/2002 (UG) und dem Satzungsteil *Studienrechtliche Bestimmungen* der Technischen Universität Wien in der jeweils geltenden Fassung. Die Struktur und Ausgestaltung des Studiums orientieren sich an folgendem Qualifikationsprofil.

§ 2 Qualifikationsprofil

Software Engineering beschäftigt sich mit der Entwicklung von Software, von der Analyse über das Design und die Implementierung bis hin zu Inbetriebnahme und Wartung.

Das Masterstudium *Software Engineering* vermittelt eine vertiefte, wissenschaftlich und methodisch hochwertige, auf dauerhaftes Wissen ausgerichtete Bildung, welche die Absolvent_innen sowohl für eine Weiterqualifizierung vor allem im Rahmen eines facheinschlägigen Doktoratsstudiums als auch für eine Beschäftigung in beispielsweise folgenden Tätigkeitsbereichen befähigt und international konkurrenzfähig macht:

- Grundlagenforschung im universitären und industriellen Bereich
- Angewandte Forschung und Entwicklung im Bereich Software in Industrieunternehmen
- Systemanalyse, Software-Architektur und Consulting

Generell haben Absolvent_innen gute Voraussetzungen für anspruchsvolle bzw. leitende Funktionen in der Wirtschaft.

Aufgrund der beruflichen Anforderungen werden im Masterstudium *Software Engineering* Qualifikationen hinsichtlich folgender Kategorien vermittelt.

Fachkompetenzen Das Studium vermittelt fortgeschrittene Kenntnisse im Bereich der Informatik und ein kritisches Verständnis ihrer Theorien und Grundsätze. Aufbauend auf einem einschlägigen Bachelorstudium werden die spezifischen Teilbereiche des Software Engineering vermittelt:

- Algorithms and Complexity
- Automation Systems and Mobile Robotics
- Data Management and Intelligent Systems
- Distributed and Next Generation Computing
- High Performance Computing
- Machine Learning
- Verification and Automated Reasoning
- Security and Privacy
- Societal Impact and Critical Reflections
- Software Engineering and Programming

Überfachliche Kompetenzen Durch die praktische und theoretische Auseinandersetzung mit aktuellen Technologien, Methoden und Werkzeugen (wie Programmiersprachen und Entwicklungsumgebungen) werden folgende Fertigkeiten vermittelt:

- Analyse und Entwurf (Requirements, Spezifikation, Modellierung)
- Realisierung von Software-Systemen und Qualitätssicherung
- Anwendung formaler Methoden, z.B. Programmanalyse, Verifikation und Model Checking
- Umgang mit Technologien, Software-Werkzeugen und Standards
- Verstehen und Verfassen von englischen Fachtexten, und Präsentieren in Englisch

Der Schwerpunkt liegt einerseits auf der Ausbildung berufsnotwendiger Zusatzkompetenzen, und andererseits auf der besonderen Förderung hoher Kreativitäts- und Innovationspotentiale. Im Einzelnen sind das:

- Kommunikation und Präsentation
- Entscheidungsverantwortung und Führungskompetenz in komplexen Projekten oder Tätigkeiten (Projektführung/Leadership)
- Gestaltungsfähigkeit

§ 3 Dauer und Umfang

Der Arbeitsaufwand für das Masterstudium *Software Engineering* beträgt 120 ECTS-Punkte. Dies entspricht einer vorgesehenen Studiendauer von 4 Semestern als Vollzeitstudium.

ECTS-Punkte (ECTS) sind ein Maß für den Arbeitsaufwand der Studierenden. Ein Studienjahr umfasst 60 ECTS-Punkte, wobei ein ECTS-Punkt 25 Arbeitsstunden entspricht (gemäß § 54 Abs. 2 UG).

§ 4 Zulassung zum Masterstudium

Die Zulassung zum Masterstudium *Software Engineering* setzt den Abschluss eines fachlich in Frage kommenden Bachelorstudiums oder eines anderen fachlich in Frage kommenden Studiums mindestens desselben hochschulischen Bildungsniveaus an einer anerkannten inländischen oder ausländischen postsekundären Bildungseinrichtung voraus. Ein Studium kommt fachlich in Frage, wenn mindestens 120 ECTS aus den Fachgebieten Mathematik und Informatk absolviert wurden, davon mindestens

1. 20 ECTS aus Mathematik und Statistik, mit vermittelten Kenntnissen, Fertigkeiten und Kompetenzen entsprechend den Modulen *Algebra und Diskrete Mathematik*, *Analysis* sowie *Statistik und Wahrscheinlichkeitstheorie* des Bachelorstudiums *Informatik* an der TU Wien;

2. 25 ECTS aus Algorithmen, Datenstrukturen und Programmierung, mit vermittelten Kenntnissen, Fertigkeiten und Kompetenzen entsprechend den Modulen *Algorithmen und Datenstrukturen*, *Datenbanksysteme*, *Programmierparadigmen*, *Einführung in die Programmierung* sowie *Software Engineering* des Bachelorstudiums *Informatik* an der TU Wien;

3. 20 ECTS aus den theoretischen, technischen und praktischen Grundlagen der Informatik, mit vermittelten Kenntnissen, Fertigkeiten und Kompetenzen entsprechend den Modulen *Denkweisen der Informatik*, *Einführung in Security*, *Grundzüge digitaler Systeme* sowie *Theoretische Informatik* des Bachelorstudiums *Informatik* an der TU Wien.

Jedenfalls ohne Ergänzungsprüfungen zuzulassen sind Absolvent_innen der Bachelorstudien *Informatik*, *Medieninformatik und Visual Computing*, *Medizinische Informatik*, *Software & Information Engineering*, *Technische Informatik* sowie *Wirtschaftsinformatik* der Technischen Universitäten Wien.

Zum Ausgleich wesentlicher fachlicher Unterschiede können Ergänzungsprüfungen im Ausmaß von maximal 30 ECTS-Punkten vorgeschrieben werden, die bis zum Ende des zweiten Semesters des Masterstudiums abzulegen sind. Diese können im Ausmaß von maximal 4,5 ECTS im Modul *Freie Wahlfächer und Transferable Skills* als freie Wahlfächer, jedoch nicht als Transferable Skills verwendet werden.

Die Unterrichtssprache ist Englisch. Studienwerber_innen, deren Erstsprache nicht Englisch ist, haben die erforderlichen Sprachkenntnisse nachzuweisen. Die Form des Nachweises ist in einer Verordnung des Rektorats festgelegt.

Manche Wahllehrveranstaltungen können auf Deutsch angeboten werden. Für diese werden Deutschkenntnisse nach Referenzniveau B1 des Gemeinsamen Europäischen Referenzrahmens für Sprachen empfohlen.

§ 5 Aufbau des Studiums

Die Inhalte und Qualifikationen des Studiums werden durch *Module* vermittelt. Ein Modul ist eine Lehr- und Lerneinheit, welche durch Eingangs- und Ausgangsqualifikationen, Inhalt, Lehr- und Lernformen, den Regelarbeitsaufwand sowie die Leistungsbeurteilung gekennzeichnet ist. Die Absolvierung von Modulen erfolgt in Form einzelner oder mehrerer inhaltlich zusammenhängender *Lehrveranstaltungen*. Thematisch ähnliche Module werden zu *Prüfungsfächern* zusammengefasst, deren Bezeichnung samt Umfang und Gesamtnote auf dem Abschlusszeugnis ausgewiesen wird.

Prüfungsfächer und zugehörige Module

Das Masterstudium *Software Engineering* gliedert sich in nachstehende Prüfungsfächer mit den ihnen zugeordneten Modulen. Die unmarkierten Module sind *Pflichtmodule* und in jedem Fall zu absolvieren. Die mit (+) markierten Module sind *Core*-Module, und

jene mit (∗) sind *Wahlmodule*. Ein Wahlmodul kann nur dann gewählt werden, wenn alle Core-Module des selben Prüfungsfachs gewählt werden.

Insgesamt sind in den Pflicht-, Core- und Wahlmodulen mit Ausnahme des Moduls *Freie Wahlfächer und Transferable Skills* Lehrveranstaltungen im Umfang von mindestens 81 ECTS zu absolvieren. Im Modul *Freie Wahlfächer und Transferable Skills* sind so viele Lehrveranstaltungen zu absolvieren, dass ihr Umfang zusammen mit der Diplomarbeit sowie dem Umfang der in den übrigen Pflicht-, Core- und Wahlmodulen gewählten Lehrveranstaltungen 120 ECTS oder mehr ergibt. Werden in den übrigen Pflicht-, Core- und Wahlmodulen insgesamt mehr als 81 ECTS absolviert, können im Modul *Freie Wahlfächer und Transferable Skills* im gleichen Ausmaß weniger ECTS absolviert werden, jedoch sind darin mindestens 4,5 ECTS aus dem Bereich der Transferable Skills zu absolvieren.

Algorithms and Complexity

+Algorithmics (6,0 ECTS)

∗Advanced Research in Algorithmics (6,0 ECTS)
∗Algorithmic Encoding Techniques (6,0 ECTS)
∗Algorithmic Geometry (6,0 ECTS)
∗Algorithmic Social Choice (6,0 ECTS)
∗Algorithms in Graph Theory (6,0 ECTS)
∗Beyond Exact Algorithms (6,0 ECTS)
∗Complexity Theory (6,0 ECTS)
∗Fixed-Parameter Algorithms and Complexity (6,0 ECTS)
∗Graph Drawing Algorithms (6,0 ECTS)
∗Heuristic Optimization Techniques (6,0 ECTS)
∗Mathematical Programming and Optimization in Transport Logistics (6,0 ECTS)
∗Structural Decompositions and Meta Theorems (6,0 ECTS)
∗Advanced Topics In Algorithms and Complexity (min. 3,0 ECTS)

Automation Systems and Mobile Robotics

+Mobile Robotics (6,0 ECTS)

∗Autonomous Racing Cars (6,0 ECTS)
∗Information Technology in Automation (6,0 ECTS)
∗Programming Principles of Mobile Robotics (6,0 ECTS)
∗Advanced Topics In Automation and Mobile Robotics (min. 3,0 ECTS)

Data Management and Intelligent Systems

+Advanced Database Systems (6,0 ECTS)

∗Business Intelligence (6,0 ECTS)
∗Database Theory (6,0 ECTS)

∗Data Stewardship (6,0 ECTS)
∗Knowledge Graphs (6,0 ECTS)
∗Logic-based Artificial Intelligence (6,0 ECTS)
∗Management of Graph Data (6,0 ECTS)
∗Problem Solving and Search in Artificial Intelligence (6,0 ECTS)
∗Processing of Declarative Knowledge (6,0 ECTS)
∗Theory of Graph Data (6,0 ECTS)
∗Advanced Topics In Data Management and Intelligent Systems (min. 3,0 ECTS)

Distributed and Next Generation Computing

+Advanced Internet Computing (6,0 ECTS)
+Distributed Systems Technologies (6,0 ECTS)
∗Artifact-based Design (6,0 ECTS)
∗Hybrid Quantum - Classical Systems (6,0 ECTS)
∗Internet of Things (6,0 ECTS)
∗Quantum Computing (6,0 ECTS)
∗Advanced Topics In Distributed and Next Generation Computing (min. 3,0 ECTS)

High Performance Computing

∗Advanced Multiprocessor Programming (6,0 ECTS)
∗GPU Computing and Architectures (6,0 ECTS)
∗Green HPC (6,0 ECTS)
∗High Performance Computing (6,0 ECTS)
∗HPC for AI (6,0 ECTS)
∗Advanced Topics In High Performance Computing (min. 3,0 ECTS)

Machine Learning

+Machine Learning (6,0 ECTS)
∗Advanced Reinforcement Learning (6,0 ECTS)
∗Algorithms for Data Science (6,0 ECTS)
∗Applied Generative AI and LLM-based Systems (6,0 ECTS)
∗Deep Learning for Natural Language Processing (6,0 ECTS)
∗Generative AI (6,0 ECTS)
∗Machine Learning for Optimization (6,0 ECTS)
∗Reinforcement Learning (6,0 ECTS)
∗Theoretical Foundations and Research Topics in Machine Learning (6,0 ECTS)
∗Theoretical Foundations of Deep Learning (6,0 ECTS)
∗Advanced Topics In Machine Learning (min. 3,0 ECTS)

Security and Privacy

∗Advanced Cryptography (6,0 ECTS)
∗Advanced Privacy Enhancing Technologies (6,0 ECTS)
∗Artificial Intelligence for Computer Security (6,0 ECTS)
∗Cryptocurrencies (6,0 ECTS)
∗Formal Methods for Security and Privacy (6,0 ECTS)
∗Network Security (3,0–6,0 ECTS)
∗Smart Contracts (6,0 ECTS)
∗Symmetric Cryptography (6,0 ECTS)
∗System and Application Security (6,0 ECTS)
∗Advanced Topics In Security and Privacy (min. 3,0 ECTS)

Societal Impact and Critical Reflections

∗Advanced Human-Centered AI: from concepts to implementation (6,0 ECTS)
∗AI Ethics (6,0 ECTS)
∗Critical Algorithm Studies (6,0 ECTS)
∗Computer Science Education: Advances in Research and Practice (6,0 ECTS)
∗Critical Theory of Media and Informatics (6,0 ECTS)
∗Human-agent Interaction (12,0 ECTS)
∗Introduction to Computational Sustainability (6,0 ECTS)
∗Learning Technologies and Learning Analytics (6,0 ECTS)
∗Responsible Digital Ethics (6,0 ECTS)
∗Advanced Topics In Societal Impact and Critical Reflections (min. 3,0 ECTS)

Software Engineering and Programming

Advanced Software Engineering (6,0 ECTS)
Advanced Software Engineering Project (6,0 ECTS)
∗Advanced Logic Programming (6,0 ECTS)
∗Advanced Model Engineering (6,0 ECTS)
∗AI Programming (6,0 ECTS)
∗Efficient Programs (6,0 ECTS)
∗Low-Level Programming (6,0 ECTS)
∗Model Engineering (6,0 ECTS)
∗Programming Paradigms and Languages (6,0 ECTS)
∗Type Systems (6,0 ECTS)
∗Advanced Topics In Software Engineering and Programming (min. 3,0 ECTS)

Verification and Automated Reasoning

+Formal Methods in Systems Engineering (6,0 ECTS)
∗Automata and Logic (6,0 ECTS)

∗Automated Deduction (6,0 ECTS)
∗Computer-Aided Verification (6,0 ECTS)
∗Logic and Computability (6,0 ECTS)
∗Program Analysis (6,0 ECTS)
∗SAT Algorithms, Applications and Extensions (6,0 ECTS)
∗Advanced Topics In Verification and Automated Reasoning (min. 3,0 ECTS)

Methods in Computer Science

Seminar in Computer Science (min. 3,0 ECTS)
∗Project in Computer Science (6,0–12,0 ECTS)

Extension

∗Extension (up to 12,0 ECTS)

Freie Wahlfächer und Transferable Skills

Freie Wahlfächer und Transferable Skills (9,0 ECTS)

Diplomarbeit

Siehe Abschnitt § 9.

Kurzbeschreibung der Module

Dieser Abschnitt charakterisiert die Module des Masterstudiums *Software Engineering* in Kürze. Eine ausführliche Beschreibung ist in Anhang A zu finden.

Advanced Cryptography (6,0 ECTS) Modern cryptography extends far beyond basic concepts like encryption and authentication to meet today's demands for security and privacy and this course presents such advanced topics. We cover the *provable security* paradigm, which yields rigorous security guarantees for practical cryptographic schemes. After introducing *elliptic-curve cryptography*, we discuss advanced forms of encryption. We intensively study *zero-knowledge proofs*, a general method for showing compliance while preserving privacy. We also touch on *secure multi-party computation* and *lattice-based cryptography*, which provides security against quantum computers.

Advanced Database Systems (6,0 ECTS) Databases are at the heart of virtually every software system. However, there are many different database system technologies and the choice depends on the requirements of the concrete application. In this module, various such technologies are put to work – including classical relational database systems, NoSQL systems, and systems specifically designed for distributed data processing in a cluster.

Advanced Human-Centered AI: from concepts to implementation (6,0 ECTS) This module provides foundational knowledge of user interfaces, "human in the loop" systems, and artificial intelligence, including supervised and unsupervised learning, as well as reinforcement learning. Students will explore key domains of human-AI interaction, such as recommender systems, chatbots, intelligent text entry, explainable AI, user modeling, and personalized and adaptive user interfaces.

Advanced Internet Computing (6,0 ECTS) Advanced Internet Computing covers modern computing infrastructures and methods to support complex Internet applications; those include scalable web services, integration and discovery, cloud deployments and models, elastic systems, business processes as well as other emerging topics. Especially, challenges inherent in contemporary use cases within smart and ubiquitous systems, call for advanced architectures, techniques, and methods. To this end, a particular focus will be on the computing continuum, that is the spectrum defined by Internet of Things on one side and the cloud on the other, driving major developments in cloud computing. Finally, the module will discuss cutting edge developments, challenges and opportunities for present-day and future cloud-enabled distributed Internet computing systems, including within security and privacy, intelligence, and mobile communication systems, with a particular focus on the interplay between AI/ML and edge/cloud computing.

Advanced Logic Programming (6,0 ECTS) Logic programming is well suited for solving complex problems and building intelligent systems. This module covers the most recent language developments that extend the pure monotonic core of ISO Prolog to enable higher-order facilities, meta programming and program transformation techniques.

Advanced Model Engineering (6,0 ECTS) The term model engineering comprises different approaches of model-driven software development such as model-driven architecture, domain-specific languages, software factories etc. All of these different approaches concentrate on a central model and not only on pure program code. In the context of this module, advanced concepts, methods, techniques, and tools of model-driven software engineering will be introduced including, ontology-driven conceptual modeling, multi-level modeling, LLM-assisted domain modeling, and web modeling.

Advanced Multiprocessor Programming (6,0 ECTS) Shared-memory multiprocessors consisting of a non-trivial number of independently programmable processor-cores with different types of hardware support for communication and synchronization are ubiquitous; but difficult to exploit efficiently and correctly. The Advanced Multiprocessor Programming course introduces thread models and notions of correctness and performance for multi-threaded programs and methods for supporting effective multi-threaded programming. Concretely, synchronization problems, operations and primitives are discussed in depth to provide a repertoire of ideas and methods (locks and atomic operations) for implementing efficient and correct multi-processor algorithms with guaranteed progress and performance guarantees. The interplay with the memory system is discussed and memory models and operations for more efficiently dealing with memory effects are presented. Basic lock-based and lock- and wait-free algorithms and data structures are introduced that can be used in algorithms and for the efficient support of tread models

and libraries. The relative power of synchronization constructs is examined with the purpose of implementing algorithms with wait- or lock-free progress guarantees. Through theoretical exercises and practical programming projects students will come appreciate the sometimes inherent difficulties for the correct and efficient use of shared-memory multiprocessors. The course material consists in back ground material, including a standard textbook, slides, hand-outs and articles.

Advanced Topics In Algorithms and Complexity (min. 3,0 ECTS) This module deepens knowledge in selected areas of algorithms and complexity.

Advanced Topics In Automation and Mobile Robotics (min. 3,0 ECTS) This module deepens knowledge in selected areas of automation and mobile robotics.

Advanced Topics In Data Management and Intelligent Systems (min. 3,0 ECTS) This module deepens knowledge in selected areas of data management and intelligent systems.

Advanced Topics In Distributed and Next Generation Computing (min. 3,0 ECTS) This module deepens knowledge in selected areas of distributed and next generation computing.

Advanced Topics In High Performance Computing (min. 3,0 ECTS) This module deepens knowledge in selected areas of high performance computing.

Advanced Topics In Machine Learning (min. 3,0 ECTS) This module deepens knowledge in selected areas of machine learning.

Advanced Topics In Security and Privacy (min. 3,0 ECTS) This module deepens knowledge in selected areas of security and privacy.

Advanced Topics In Societal Impact and Critical Reflections (min. 3,0 ECTS) This module deepens knowledge in selected areas of societal impact and critical reflections.

Advanced Topics In Software Engineering and Programming (min. 3,0 ECTS) This module deepens knowledge in selected areas of software engineering and programming.

Advanced Topics In Verification and Automated Reasoning (min. 3,0 ECTS) This module deepens knowledge in selected areas of verification and automated reasoning.

Advanced Privacy Enhancing Technologies (6,0 ECTS) This module equips students with state-of-the-art techniques, such as differential privacy, secret sharing, and secure multi-party computation, that enable secure data processing without compromising confidentiality. These methods allow sensitive data to be utilized in AI applications and analytics while ensuring strong privacy guarantees, addressing key concerns in sectors like healthcare, finance, and social media. By bridging the gap between privacy protection and data utility, the lecture prepares students to tackle modern challenges in balancing ethical data usage with technological advancement.

Advanced Reinforcement Learning (6,0 ECTS) This module covers advanced topics in reinforcement learning. Regarding algorithms, distributional reinforcement learning

and distributional deep reinforcement learning are discussed. Reinforcement learning plays the crucial role in the development of large language models (LLMs) such as ChatGPT and DeepSeek-R1; these algorithms are presented as well. Convergence proofs of important algorithms are given. In the tutorial part, students present the theory of recent algorithms in detail or implement learning algorithms for complex environments.

Advanced Research in Algorithmics (6,0 ECTS) This module introduces students to research-oriented work in algorithmics. Students engage with selected topics from fields such as computational social choice, algorithmic game theory, computational geometry, graph algorithms, and approximation techniques, developing skills in literature analysis, critical thinking, and research formulation.

Advanced Software Engineering (6,0 ECTS) Reliability issues in modern software often result in significant financial losses or system downtime, thereby affecting millions of users. This module discusses in detail fundamental techniques for ensuring the reliability of large, complex software systems. Such techniques are heavily used in practice and range from static program analysis and verification to automated test generation. The module also focuses on applications of these techniques to different, popular domains, such as smart contracts and machine-learning models.

Advanced Software Engineering Project (6,0 ECTS) The engineering of large software systems, e.g., open source software, business or software-intensive systems, is typically a group effort in agile or plan-driven projects. This module shall provide the framework for student groups to exercise advanced software engineering in a project, resulting in software prototypes and documentation that will typically be useful for and usable by real users.

AI Ethics (6,0 ECTS) With the rapid developments in Artificial Intelligence (AI), we find an inevitable increase in AI's impact on our lives, posing various ethical challenges for developers, policy-makers, and society at large. This has led to a general awareness that AI systems must be aligned with human values, ethics, and laws. This module introduces various ways in which ethics and AI meet. It enables students to develop basic skills for identifying and critically reflecting on AI-specific ethical challenges, and for incorporating critical thinking on these topics in their work and further studies.

AI Programming (6,0 ECTS) The development of AI systems increasingly relies on specialized programming methodologies that enable efficient and scalable implementation. This module introduces fundamental concepts and practical methodologies required for AI-driven system development. Topics include programming techniques and paradigms relevant to AI, as well as underlying techniques and models that enable AI for code. As part of this, the module explores paradigms such as probabilistic programming and differentiable programming, along with broader techniques applicable to AI development.

Algorithmic Encoding Techniques (6,0 ECTS) This module explores advanced techniques for solving complex computational problems through encoding into established target formalisms. The module focuses on methodological approaches to problem encoding, including satisfiability, maximum satisfiability, and constraint programming. Students

learn systematic encoding strategies and techniques for improving solving efficiency, with emphasis on correctness certification and handling dynamic problem instances.

Algorithmic Geometry (6,0 ECTS) Algorithmic geometry deals with efficient computations on basic geometric objects such as points, lines, polygons, and their higher-level analogs. Geometric algorithms play an important role in a variety of applications, e.g., visual computing, geographic information systems, machine learning, robotics, etc. This module studies the design and analysis of geometric algorithms and efficient data structures. We present fundamental techniques and concepts in algorithmic geometry as well as more advanced results and discuss selected geometric problems arising in applications.

Algorithmic Social Choice (6,0 ECTS) Computational social choice studies algorithmic and computational aspects of collective decision-making processes, including voting, fair division, and matching problems. It provides fundamental tools for addressing fairness, efficiency, and strategic behavior in group decisions. Applications arise in political elections, resource allocation, multi-agent systems, and online platforms. This module introduces key concepts in computational social choice, covering preference aggregation, complexity analysis, and algorithmic techniques. We study fundamental problems as well as selected advanced topics, with emphasis on both theoretical foundations and algorithmic solutions.

Algorithmics (6,0 ECTS) This module thoroughly treats advanced algorithms and data structures, emphasizing theoretical foundations and algorithm analysis. The module focuses on mathematical methods for analyzing computational problems and developing efficient algorithmic solutions. Students learn to understand and prove the correctness and efficiency of algorithms through formal mathematical methods.

Algorithms for Data Science (6,0 ECTS) This module introduces fundamental algorithms for analyzing large amounts of data. It is based on important results of the algorithms community which are widely recognized for their practical applications in data science. The module will cover algorithms for questions, such as: How can one find duplicate pages on the web? How can one detect communities in social networks? How can we embed high-dimensional data into low-dimensional spaces, to analyze data more efficiently? The module's focus will be on the underlying algorithmic principles and the theoretical algorithm analysis, i.e., we will prove that these algorithms work and why. We will also (to a lesser extent) consider use-cases of these algorithms in practice.

Algorithms in Graph Theory (6,0 ECTS) This module provides a thorough theoretical treatment of graph algorithms and their mathematical foundations. The module focuses on fundamental concepts in graph theory and their algorithmic aspects. Students learn to develop and analyze algorithmic solutions through formal mathematical methods.

Applied Generative AI and LLM-based Systems (6,0 ECTS) This module focuses on the practical implementation and integration of generative AI systems. It introduces modern model architectures such as mixture-of-experts, advanced optimization techniques including parameter-efficient tuning, and integration patterns such as tool use, function calling, or agentic systems. Through practical group projects, this module aims to transfer theoretical understanding to practical implementations of generative AI systems.

Artifact-based Design (6,0 ECTS) By engaging with hands-on prototyping, testing, and iterative design, students will develop skills in designing and implementing tangible and digital artifacts that reflect user needs and societal contexts. Through lectures, collaborative projects, and reflective critiques, participants will learn to develop artifacts as tools for understanding problems, communicating ideas, and evaluating concepts. This module emphasizes a practical, research-oriented approach to design, enabling students to translate theoretical knowledge into meaningful and impactful integrated physical-digital concepts.

Artificial Intelligence for Computer Security (6,0 ECTS) This module explores how artificial intelligence and machine learning can address critical security challenges, such as malware detection and vulnerability discovery, that are no longer scalable through manual efforts. Participants will learn how ML techniques can improve security while also examining the new attack vectors introduced by these models. Key topics include leveraging explainable AI to analyze and mitigate risks in AI-driven systems.

Automata and Logic (6,0 ECTS) This module provides an introduction to the interplay between automata theory, logic, and algebra, three foundational pillars of theoretical computer science. Students will explore how automata serve as computational models for languages and systems, how logic provides precise tools for specifying and reasoning about them, and how algebraic methods unify and deepen our understanding of both.

Automated Deduction (6,0 ECTS) The reasoning power that computational logic offers brings new perspectives in the field of system verification and certification. This module is about computational logic, with particular focus on algorithmic and automated methods for proving logical properties expressed in first-order logic with equality. Such properties naturally model safety/security assumptions on computer systems; for example memory accesses are naturally modeled using unbounded data structures, and hence quantifiers. The module aims at teaching attendees algorithmic techniques and fundamental results in first-order theorem proving for logic with equality. We will address both the theoretical and practical aspects for using and implementing reasoning engines for such logic.

Autonomous Racing Cars (6,0 ECTS) The goal of this module is to give students an up-to-date foundation in the technologies being deployed and tested on self-driving cars, and more general mobile autonomous systems. This hands-on, lab-centered module is for master students interested in the fields of artificial perception, motion planning, control theory, and applied machine learning. It is also for students interested in the burgeoning field of autonomous driving. Every week, (pre-recorded) lectures covering the relevant aspects are provided. Core of the lecture are the lab assignments. They are starting with the hard- and software that is needed to build and programme a 1/10th scale autonomous race car. Then fundamental principles in perception, planning and control and map-based approaches follow. At the end students develop and implement advanced racing strategies, computer vision and machine learning algorithms that will give their team the edge in the race that concludes the module.

Beyond Exact Algorithms (6,0 ECTS) Exact algorithms always give the correct answer or provide optimal solutions but they can be complex and computationally intensive. This module explores alternatives to exact algorithms through the study of randomized and approximation algorithms. Such algorithms can be significantly simpler and more efficient by either making controlled random decisions during execution or finding near-optimal solutions. The module covers design principles and analysis methods for both paradigms, including probabilistic performance guarantees, approximation ratios, and running-time bounds.

Business Intelligence (6,0 ECTS) In today's data-driven world, Business Intelligence (BI) plays a crucial role in transforming raw data into actionable insights that support strategic decision-making. This module provides a comprehensive introduction to BI principles, methodologies, and technologies. Students will explore key topics in data warehousing and data mining, covering OLAP and multi-dimensional data modeling as well as data analytics process models and regulatory principles.

Complexity Theory (6,0 ECTS) When developing algorithms for different problems, it turns out that some problems admit efficient algorithms while others don't. Complexity theory provides tools that allow us to explain why for a given problem no "better" algorithm can be expected, where "better" typically means faster or with less memory consumption. It may also mean that a problem is better suited for parallel processing than others.

Computer-Aided Verification (6,0 ECTS) This module focuses on the automated verification of hardware and software systems using model checking techniques. It covers specification languages for system properties, system models, and verification algorithms.

Computer Science Education: Advances in Research and Practice (6,0 ECTS) The module "Computer Science Education: Advances in Research and Practice"provides a deeper exploration of concepts and methods in computer science education practice and research. In an interactive course interested students receive the opportunity to delve into current research topics in computer science education and work on their own research and development project within the eduLAB context. As part of the course, students design and implement their own computer science education interventions, applying research methods in a practical setting and developing tasks, activities, or learning materials based on current insights.

Critical Algorithm Studies (6,0 ECTS) This module takes an interdisciplinary approach to examining the ethical, social, and political dynamics of algorithmic systems. It explores how algorithms influence society, focusing on issues like bias, fairness, transparency, accountability, and trust and safety. Through theoretical and practical discussions, students critically assess the harms and safeguards of algorithms in real-world settings, tackling challenges such as misinformation, extremist content, and online harassment. The module also includes a final report where students design technical or policy-based solutions to mitigate algorithmic harms.

Critical Theory of Media and Informatics (6,0 ECTS) This module enables students to develop an understanding of technology and media in society through a critical

theory lens. A philosophical foundation is provided, discussing a wide range of thinkers and their central ideas. Students will be able to discuss cutting edge trends in digital technology and media along perspectives such as power, oppression, meaning, culture and the political dimensions of technology. They will further learn how to translate such critical thinking into a future oriented design practice to develop alternative technological futures.

Cryptocurrencies (6,0 ECTS) The module provides in-depth knowledge on the foundations of blockchain technologies.

Data Stewardship (6,0 ECTS) Data plays an increasingly important role, both in research as well as in industry applications. This module focuses on data and its processing, ensuring that it meets quality requirements of the respective types of use. This module further provides knowledge on the life cycle of data, from data collection to long-term provision and reuse as well as the correct management of sensitive data.

Database Theory (6,0 ECTS) This module delves into the theoretical foundations of databases, emphasizing the expressive power, computational complexity, and formal properties of query languages and data models. Students will explore topics such as the semantics of query languages, reasoning with integrity constraints, decidability, and optimization from a mathematical perspective.

Deep Learning for Natural Language Processing (6,0 ECTS) This module provides an in-depth introduction to deep learning for natural language processing (NLP). Students will explore fundamental and advanced deep learning techniques for NLP applications, including theoretical concepts and practical implementations using PyTorch.

Distributed Systems Technologies (6,0 ECTS) The module aims to implement distributed enterprise applications using appropriate modern distributed systems technologies. The focus is on understanding the theory and concepts underlying these technologies and, therefore, relating other or new technologies to corresponding problems in distributed systems (e.g., remoting, distributed transactions, caching, API descriptions, messaging, monitoring, or auto-scaling). Students are, therefore, able to make informed decisions about which technologies to use during the design and development phases of distributed enterprise applications.

Efficient Programs (6,0 ECTS) Inefficient programs result in increased hardware costs, or, in some cases, in the software project being canceled. This module discusses the role of efficiency in software, various forms of efficiency, the resource costs of various operations and various data, transformations for increasing efficiency, and the use of tools for understanding software efficiency aspects. The module also gives examples of improving the efficiency of software.

Extension (up to 12,0 ECTS) This module allows students to extend their profile either by choosing courses from other Master curricula that fit the qualification profile of *Software Engineering*.

Fixed-Parameter Algorithms and Complexity (6,0 ECTS) Fixed-parameter algorithms provide a powerful approach for efficiently solving many NP-hard problems by exploiting structural aspects of problem instances in terms of a problem parameter. This module provides an overview of the main techniques for developing fixed-parameter algorithms as well as the fundamentals of parameterized complexity theory.

Formal Methods for Security and Privacy (6,0 ECTS) The module provides in-depth knowledge on the formal methods used in industry and academia to certify the security and privacy of software, cryptographic protocols, blockchains, machine learning, and more.

Formal Methods in Systems Engineering (6,0 ECTS) Ensuring system reliability became mandatory in our digital world. The area of formal methods provide rigorous arguments to prove that systems have no errors and behave as expected. Yet, there are theoretical results showing that there is no "one" formal approach that can be used for every system error, in every technology. This module focuses for software and hardware systems and presents formal methods for ensuring trustworthiness of such systems. We address SAT/SMT solving, deductive program verification, model checking, and static code analysis. Module concepts will be demonstrated using fully automated tools, allowing students to practice and consolidate their knowledge in formal methods for efficient systems engineering.

Freie Wahlfächer und Transferable Skills (9,0 ECTS) Die Lehrveranstaltungen dieses Moduls dienen der Vertiefung des Faches sowie der Aneignung außerfachlicher Kenntnisse, Fähigkeiten und Kompetenzen.

Generative AI (6,0 ECTS) Generative AI has brought fundamental paradigm shifts in AI and beyond. This module provides a comprehensive introduction to both theoretical foundations and practical applications of it. The module explores principles of generative AI around language modeling, image generation, multimodal systems and discusses intersections between generative AI and other research areas such as knowledge-based architectures and recommender systems. Besides theoretical foundations, it covers practical implementation aspects such as prompt engineering, retrieval-augmented generation, and modern tooling/frameworks. Finally, the module also addresses crucial ethical considerations and societal implications of generative AI systems.

GPU Computing and Architectures (6,0 ECTS) In today's data-driven world, Graphics Processing Units (GPUs) have emerged as indispensable tools for accelerating computationally intensive tasks in areas such as scientific simulation, image processing, and machine learning. This module, GPU Computing and Architectures, introduces students to the fundamentals of GPU hardware design and parallel programming principles, with a focus on NVIDIA's CUDA platform. Throughout the module, you will explore essential GPU topics, including memory organization, efficient parallel patterns, and advanced features like dynamic parallelism. Hands-on projects and collaborative assignments will enable you to apply theoretical concepts to real-world applications, enhance your problem-solving skills, and develop effective teamwork strategies. By the

end of the module, you will be equipped to design and optimize high-performance parallel algorithms on modern many-core processors.

Graph Drawing Algorithms (6,0 ECTS) Graph drawing is concerned with the geometric representation of graphs in the plane and constitutes the algorithmic core of network visualization. The research area of graph drawing combines aspects of algorithmics, graph theory, computational geometry, and visualization. In this module we define common aesthetic quality criteria and layout styles in graph drawing. Subsequently, we study the corresponding optimization problems from a formal, algorithmic perspective. We cover some of the most fundamental graph drawing algorithms, ranging from general-purpose algorithms to specific algorithms for certain graph classes (e.g., trees and planar graphs). The algorithms use known algorithm design principles such as divide-and-conquer, incremental constructions, and network flow models. The module covers both practical and theoretical aspects of graph drawing.

Green HPC (6,0 ECTS) As HPC systems increasingly power breakthroughs in AI, scientific research, and engineering, their significant energy consumption and environmental impact have become critical challenges. This module addresses these issues by introducing core concepts, sustainability measures, and advanced techniques for reducing the carbon footprint of HPC systems. This module deals with the basic energy impact of HPC, sustainability measures and key performance indicators (KPIs), methods and tools for assessing energy consumption and CO2 footprints, and green workload scheduling algorithms.

Heuristic Optimization Techniques (6,0 ECTS) This module deals with algorithmic techniques to heuristically solve practicably challenging optimization problems in reasonable time.

High Performance Computing (6,0 ECTS) High-Performance Computing (HPC) aims to provide means and methods to achieve the highest possible performance out of given, typically very large, parallel computing systems. This module will provide an overview of current HPC architectures and communication networks, and of problems, algorithms and solutions for different application areas, including tools and libraries. It covers current programming paradigms in detail at an advanced level, notably MPI but also other, related frameworks, and will go into algorithmic and engineering aspects of the efficient implementation of frameworks and libraries for HPC. Through projects and exercises students will appreciate challenges and difficulties in achieving and substantiating claims of high performance for selected problems and architectures.

HPC for AI (6,0 ECTS) Artificial Intelligence (AI) in High-Performance Computing (HPC) faces several challenges, primarily driven by rapid hardware advancements and the explosion of data. As AI applications become increasingly complex and data-intensive, HPC systems must evolve to handle the growing demand for computational power while managing energy consumption and maintaining scalability. This requires innovative methods to model, implement, and verify tradeoffs between accuracy and performance, ensuring that AI workloads in HPC achieve the desired balance between computational efficiency and result precision. In this lecture students will learn fundamentals for the

development of advanced algorithms and techniques that ensure the robustness and scalability of AI workflows across diverse HPC architectures.

Human-agent Interaction (12,0 ECTS) This module explores the research field of Human-Agent Interaction (HAI) from a human-centered perspective. It examines the socio-technical complexities of interacting with systems with apparent agency, such as robots, chatbots, and other intelligent systems. Students explore how humans perceive and interact with these agents, the design of engaging and effective interactions, and evaluation methods from a human-centered perspective. Topics include embodiment, anthropomorphism, socio-morphism, agent design, multimodal communication, and real-world applications. The module integrates lectures, critical readings, discussions, and practical user research techniques.

Hybrid Quantum - Classical Systems (6,0 ECTS)
As we enter the post-Moore era, new computing architectures are emerging, with quantum computing standing out as one of the most promising advancements. However, to unlock the full potential of quantum machines, they must interact effectively with classical systems. Hybrid classical-quantum systems enable this integration, ensuring the efficient use of quantum resources, the practical application of quantum algorithms, and seamless compatibility with existing classical infrastructures. These systems represent a vital step forward in shaping the future of computing.

Information Technology in Automation (6,0 ECTS) The interplay of ubiquitous computing, fog/cloud infrastructure, the Internet of Things (IoT) and various services opens up new opportunities for the application of automation technology, but also poses new quantitative and qualitative challenges in terms of their safe and secure integration into standardized industrial system architectures. The module aims to provide students with an in-depth understanding of the topics regarding the vertical and horizontal integration of devices and services in an distributed automation environment.

Internet of Things (6,0 ECTS) This module aims to equip the students with the necessary engineering and scientific competencies to design Internet of Things solutions that enable everyday objects to collect, send and receive data through the internet.

Introduction to Computational Sustainability (6,0 ECTS) This lecture addresses the urgent need to align the rapid growth of technology and computing with the pressing challenges of environmental, societal, and economic sustainability.

Knowledge Graphs (6,0 ECTS) Knowledge graphs combine scalable graph data management with both symbolic AI (e.g., logic-based) and subsymbolic AI (i.e., machine learning). Key techniques are logic-based reasoning, knowledge graph embeddings, graph neural networks, and graph transformers. An overarching aim of the module is to understand the connections between knowledge graphs, artificial intelligence, neuro-symbolic AI, Machine Learning, Deep Learning and Data Science.

Learning Technologies and Learning Analytics (6,0 ECTS) This module introduces the research fields of learning technologies and learning analytics: Learning technologies is an application area of computer science that deals with educational systems and learning

processes. Learning analytics utilizes data on learning processes to gain insights and support users. Both involve methods of human-centered design, software engineering, and data science. The module consists of a lecture introducing different concepts and technologies for teaching and learning, and a integrated practical assignments and a group project on designing and implementing a prototype for learning analytics.

Logic and Computability (6,0 ECTS) The module aims at a consolidation as well as an extension of skills and knowledge in mathematical logic and computability theory acquired at the bachelor level. It provides advanced training in using logical formalisms as tools for specification and problem-solving across diverse applications. Among other things, students will study various logical systems for proof search, explore elements of modal and intuitionistic logic, and gain a deeper understanding of computability, including its connections to logic, program verification, and complexity theory.

Logic-based Artificial Intelligence (6,0 ECTS) This module provides an advanced introduction to logic-based artificial intelligence. The students will learn selected logic formalisms used to represent complex domains of interest and the computational methods for deriving conclusions from these representations. By discussing both theoretical principles and practical techniques, the module aims to prepare students for advanced applications and research in AI. The module begins by introducing predicate logic as a specification language, and then moves on to description logics (DL) for representing ontological knowledge. The students will learn the basics of DL reasoning; special attention will be given to the description logic EL popular in biomedical informatics. The students will learn the principles behind nonmonotonic reasoning, and become acquainted with answer-set programming and its application for declarative problem solving in combinatorial domains. Students will learn the foundations of probabilistic reasoning and, finally, rule learning, bridging the gap between logical inference and data-driven learning. By the end of the module, students will have developed both theoretical expertise and practical skills, enabling them to deploy logic-based AI solutions to solve problems in different domains.

Low-Level Programming (6,0 ECTS) While most application software is written in high-level programming languages, at the bottom of the software stack there is a foundation of low-level programming. This module discusses the reasons for low-level programming and its problems in security and reliability. It also provides competencies in low-level programming, including programming languages and debugging.

Machine Learning (6,0 ECTS) The module provides a broad introduction to fundamental topics in machine learning. It covers core concepts of machine learning, the basics of machine learning theory, and the main techniques in supervised learning, unsupervised learning, and reinforcement learning.

Machine Learning for Optimization (6,0 ECTS) Combinatorial optimization problems arise in various domains, and solving them is crucial to improve efficiency and resource consumption. Traditionally, such problems have been solved using exact methods and heuristic techniques. In recent years, machine learning approaches have been developed and used to improve traditional solvers and to directly solve such problems.

This module provides an introduction to methods for solving combinatorial optimization problems based on machine learning. It includes various learning approaches based on supervised learning, reinforcement learning, and large language models for hard combinatorial optimization problems.

Management of Graph Data (6,0 ECTS) Graph data, such as knowledge graphs, plays a pivotal role in numerous cutting-edge applications, including graph analytics, data integration, and machine learning. However, managing graph data effectively requires navigating a diverse landscape of technologies and selecting the right methods to meet specific application requirements. This module provides a comprehensive introduction to the principles, technologies, and advanced topics in graph data management. Students will explore popular graph data models, query languages, and management systems, covering topics such as query formulation, query optimization, schema design, provenance, data integration, and dynamic updates. By completing the module, participants will be well-equipped to apply graph data technologies in real-world scenarios and engage with emerging trends in this rapidly evolving field.

Mathematical Programming and Optimization in Transport Logistics (6,0 ECTS) This module covers the theory and practice of solving challenging optimization problems by means of mathematical programming techniques, in particular integer linear programming methods. While highly relevant in many application areas, a particular focus is put on optimization problems in transport logistics.

Mobile Robotics (6,0 ECTS) This module covers fundamental concepts and techniques in mobile robotics. It explores the key challenges of autonomous intelligent systems, focusing on state-of-the-art solutions for path and motion planning, self-localization, and SLAM. Practical exercises are conducted using both simulated and real robotic hardware.

Model Engineering (6,0 ECTS) The term model engineering comprises different approaches of model-driven software development such as model-driven architecture, domain-specific languages, software factories etc. All of these different approaches concentrate on a central model and not only on pure program code. During this module the different concepts, tools, and practical experiences from the field of model engineering will be examined. The module is split up into a theoretical part and an accompanying model engineering lab. Concepts from the field of metamodeling, development of textual and graphical modeling languages, model transformation, and code generation will be taught in the lecture part. At the beginning of the lab students will form teams. During the labs these teams will be given practical assignments chosen from the topics of the lecture part.

Network Security (3,0–6,0 ECTS) The module comprises two courses. The course *Network Security* covers network security basics, attack techniques, cryptography basics and selected methods (RSA, ElGamal, Diffie-Hellman, Elliptic Curve Cryptography), network traffic analysis and anomaly detection. In the practical exercises students learn techniques to find attacks and attack preparation activities. The course *Network Security - Advanced Topics* covers advanced topics in network security such as IPv6 security, routing security, secure group communication, network steganography (hidden communication in

covert and subliminal channels) and future challenges (such as smart grid security). In the practical exercises students learn methods to detect hidden communication in network traffic. Both parts contain network traffic analysis exercises and therefore students need to apply statistical data analysis and anomaly detection methods.

Problem Solving and Search in Artificial Intelligence (6,0 ECTS) Solving logical problems and complex real-world constraint satisfaction and optimization problems requires strong problem solving skills. While experienced human experts can often find good solutions for small problems, efficiently solving large-scale problems and automating problem-solving are important goals of AI. This module covers core topics in AI problem solving including problem modeling, AI-based exact and heuristic search techniques, learning in search, and hybrid approaches. It also demonstrates the application of these techniques in various practical domains, including planning and scheduling.

Processing of Declarative Knowledge (6,0 ECTS) This module introduces methods and techniques for declarative knowledge processing. In this paradigm, knowledge and problem specifications can be expressed using sentences in symbolic languages, while specialized reasoning engines can be used to process these sentences. Unlike procedural approaches, which require programmers to explicitly define how computations should be performed, declarative methods allow reasoning systems to determine the necessary computations based on the input specification. This distinction makes declarative methods a powerful alternative to procedural approaches. By separating problem specifications and domain knowledge from implementation details, declarative methods offer benefits such as improved solution quality, reusability, transparency, and explicability.

Program Analysis (6,0 ECTS) This module provides an in-depth introduction to program analysis, a set of techniques and tools for automatically reasoning about the behavior and properties of software programs. We will explore foundational concepts, including static analysis, program semantics, abstract interpretation, and type systems, along with their applications in verification, optimization, and software quality.

Programming Paradigms and Languages (6,0 ECTS) Programming paradigms as well as programming languages tremendously influence our way of thinking when developing software. In this module we investigate the design space of programming languages and examine how language design decisions influence paradigms and the quality of developed software. We highlight strengths and pitfalls of major paradigms and show how to deal with them. The primary goal is to get better awareness of mutual interactions between paradigms, languages and our thinking.

Programming Principles of Mobile Robotics (6,0 ECTS) This module covers state-of-the-art algorithms and methodologies from the domain of autonomous as much as semi-autonomous robotic systems like self-driving cars, drones, or search-and-rescue robots. Therefore, selected topics from the domain of software architecture and frameworks, artificial intelligence and cognitive robotics, and cyber-physical-systems are discussed, and will also be implemented and applied to specialized robotic hardware by students.

Project in Computer Science (6,0–12,0 ECTS) In this module, practical problems from the field of Computer Science are solved. This gives insights into scientific practice

and current research in Computer Science. To tackle larger problems, two practical projects can be combined.

Quantum Computing (6,0 ECTS) This module provides the fundamental concepts of quantum computers and discusses both basic as well as advanced quantum algorithms and quantum complexity classes. Moreover, it covers issues related to quantum information, cryptography, and quantum teleportation. Necessary preliminaries from mathematics and quantum mechanics are provided also.

Reinforcement Learning (6,0 ECTS) Reinforcement learning is the field of machine learning that is concerned with the control of time-dependent systems. It can also be described as stochastic optimal control, or as dynamic programming plus learning. It can be used with or without models. In this module, the basic data structures such as Markov decision processes are defined, and the most important learning algorithms are derived. Furthermore, important theoretic results regarding convergence to optimal solutions are discussed. In the last part of the class, deep reinforcement learning and the algorithms that were developed to solve chess, Go, shogi, the Atari 2600 games, and further computer games are presented. In the tutorial part, algorithms are implemented, sample problems are solved, and theoretic questions are answered.

Responsible Digital Ethics (6,0 ECTS) This module addresses the critical importance of responsible digital ethics in the age of pervasive digital systems and increasing regulatory requirements. Students explore foundational concepts such as trust, fairness, accountability, digital rights, and ethical design principles. Through lectures, interactive seminars, and student-led activities, participants engage with real-world challenges and diverse literature to develop a nuanced, experiential understanding of digital ethics and their role in fostering ethical digital practices.

SAT Algorithms, Applications and Extensions (6,0 ECTS) Formal verification techniques for software and hardware systems often rely on Boolean satisfiability (SAT) solvers, which have already become powerful practical tools for solving a wide range of problems in formal methods, artificial intelligence, and beyond. This module provides a deep dive into the algorithms and advanced data structures behind SAT solvers, some of their practical applications, and their extensions to reasoning beyond propositional logic, such as solvers for SMT (satisfiability modulo theories), MaxSAT (maximum satisfiability), and QBF (quantified Boolean formulas). Through a combination of lectures, exercises, and small programming projects, students will gain both theoretical insights and hands-on experience with state-of-the-art tools. Topics include core SAT solving techniques (such as the CDCL algorithm combined with formula simplification techniques and incremental reasoning), advanced heuristics, and practical applications. By the end of the module, students will not only understand how SAT and SAT-based solvers work but also how to apply and extend them to solve new challenging problems.

Seminar in Computer Science (min. 3,0 ECTS) In the seminar, students dive deeper into the literature of a chosen topic in Computer Science and present their findings in written and oral form.

Smart Contracts (6,0 ECTS) Smart contracts are cryptocurrency-based, event-driven computer programs that run on a peer-to-peer network, the blockchain, with the purpose to automate the exchange of digital assets without the need for an external trusted authority. This module focuses on blockchains as a distributed programming platform. It presents the peculiarities of smart contracts, common patterns and pitfalls, security aspects, as well as the methods and technologies for developing secure smart contracts.

Structural Decompositions and Meta Theorems (6,0 ECTS) This module explores the relationship between structural decompositions and logical frameworks in algorithmic problem-solving. The module develops two complementary perspectives: structural parameters that measure the complexity of input instances and logical frameworks that can express families of algorithmic problems. These come together in meta-theorems that establish general conditions for efficient solvability.

Symmetric Cryptography (6,0 ECTS) Symmetric cryptography studies the cryptographic algorithms that secure the bulk of digital data. That includes the AES blockciphers (used in D/TLS, smart cards, on-chip security, bank payments, etc.), authenticated/encryption mechanisms like CTR, CBC, GCM, and CCM (used in the D/TLS protocol for secure Internet, pseudorandom number generation, storage encryption, etc.), hash functions like SHA-2 and SHA-3 (used for digital signing, D/TLS, authentication tokens, password protection, etc.), and novel building blocks (MiMC hash, Poseidon hash, Merkle trees) that are used in privacy-friendly blockchain and cloud computation protocols. In this module, we examine the security rationale for all those designs both from a theory and a practical perspective. Furthermore, we demonstrate attack vulnerabilities and reason security in the provable security theoretical framework.

System and Application Security (6,0 ECTS) This module covers security of software in different scenarios and at different system levels. Students learn to find and analyze security vulnerabilities in complex computer systems, as well as to explain the root causes of identified vulnerabilities and develop mitigations.

Theoretical Foundations and Research Topics in Machine Learning (6,0 ECTS) Machine learning enables computers to learn from data and make intelligent decisions. Through a combination of theoretical analysis and practical applications, this module provides a rigorous introduction to the theoretical aspects of machine learning. Students will develop the ability to prove key results in learning theory, analyze and compare machine learning algorithms, and apply them effectively. Additionally, the module will enhance the students' ability to critically engage with the machine learning literature and present some of the latest developments in the field.

Theoretical Foundations of Deep Learning (6,0 ECTS) Deep learning surpasses human accuracy in classification tasks and outperforms champions in games like Go. As its applications grow, practitioners have identified key properties behind its effectiveness, such as early layers resembling dictionary learning, and deeper layers encoding invariance to transformations. This module explores emerging mathematical theories that explain these observations.

Theory of Graph Data (6,0 ECTS) Graph-structured data is becoming an increasingly prominent method for storing and then providing sophisticated access methods for information with complex structure, especially in scenarios where it is difficult to come up with a strict database schema required in traditional relational databases. This data model is at the core of recent technologies likes knowledge graphs, property graphs, RDF graphs, and graph databases. Modeling and design, integrity maintenance, as well as querying answering in graph-structured data are quite different from the analogous tasks for standard relational database, which is due to the different use-cases and requirements in the context of graph-structured data. The goal of this module is to provide a solid understanding of the foundations of various methods and techniques for graph-structured data, which will enable the students to compare their uses, limits and possibilities. Studying these foundations through the lens of database theory, computational logic, and computational complexity, the students will achieve an understanding that abstracts from the concrete existing implementations and official standards, enabling them to effectively leverage the potential of existing technologies.

Type Systems (6,0 ECTS) While programming languages depend on different kinds of types as mechanisms to ensure compatibility between operators and operands, software developers use these types as basic tools for building abstractions with well-defined properties. Type systems are rule systems specifying relationships between expressions and types. In this module we discuss type systems from the language perspective, always keeping the requirements of software developers in mind. In addition to formal foundations the focus is on type systems used in functional and object-oriented programming.

§ 6 Lehrveranstaltungen

Die Stoffgebiete der Module werden durch Lehrveranstaltungen vermittelt. Die Lehrveranstaltungen der einzelnen Module sind in Anhang A in den jeweiligen Modulbeschreibungen spezifiziert. Lehrveranstaltungen werden durch Prüfungen im Sinne des Universitätsgesetzes beurteilt. Die Arten der Lehrveranstaltungsbeurteilungen sind in der Prüfungsordnung (§ 7) festgelegt.

Betreffend die Möglichkeiten der Studienkommission, Module um Lehrveranstaltungen für ein Semester zu erweitern, und des Studienrechtlichen Organs, Lehrveranstaltungen individuell für einzelne Studierende Wahlmodulen zuzuordnen, wird auf § 27 des studienrechtlichen Teils der Satzung der TU Wien verwiesen.

Vorgaben zu Lehrveranstaltungen und Prüfungen aus dem Universitätsgesetz 2002

Vor Beginn jedes Semesters ist ein elektronisches Verzeichnis der Lehrveranstaltungen zu veröffentlichen (Titel, Name der Leiterin oder des Leiters, Art, Form inklusive Angabe des Ortes und Termine der Lehrveranstaltung). Dieses ist laufend zu aktualisieren.

Die Leiterinnen und Leiter einer Lehrveranstaltung haben, zusätzlich zum veröffentlichten Verzeichnis, vor Beginn jedes Semesters die Studierenden in geeigneter Weise über

die Ziele, die Form, die Inhalte, die Termine und die Methoden ihrer Lehrveranstaltungen sowie über die Inhalte, die Form, die Methoden, die Termine, die Beurteilungskriterien und die Beurteilungsmaßstäbe der Prüfungen zu informieren.

Für Prüfungen, die in Form eines einzigen Prüfungsvorganges durchgeführt werden, sind Prüfungstermine jedenfalls drei Mal in jedem Semester (laut Satzung am Anfang, zu Mitte und am Ende) anzusetzen, wobei die Studierenden vor Beginn jedes Semesters über die Inhalte, die Form, die Methoden, die Termine, die Beurteilungskriterien und die Beurteilungsmaßstäbe der Prüfungen zu informieren sind.

Bei Prüfungen mit Mitteln der elektronischen Kommunikation ist eine ordnungsgemäße Durchführung der Prüfung zu gewährleisten, wobei zusätzlich zu den allgemeinen Regelungen zu Prüfungen folgende Mindesterfordernisse einzuhalten sind:

- Vor Semesterbeginn Bekanntgabe der Standards, die die technischen Geräte der Studierenden erfüllen müssen, damit Studierende an diesen Prüfungen teilnehmen können.
- Zur Gewährleistung der eigenständigen Erbringung der Prüfungsleistung durch die Studierende oder den Studierenden sind technische oder organisatorische Maßnahmen vorzusehen.
- Bei technischen Problemen, die ohne Verschulden der oder des Studierenden auftreten, ist die Prüfung abzubrechen und nicht auf die zulässige Zahl der Prüfungsantritte anzurechnen.

Vorgaben zu Lehrveranstaltungen aus der Satzung der TU Wien

Im Folgenden steht SSB für *Satzung der TU Wien, Studienrechtliche Bestimmungen.*

- Der Umfang einer Lehrveranstaltung ist in ECTS-Anrechnungspunkten und in Semesterstunden anzugeben. [§ 9 SSB (Module und Lehrveranstaltungen)]
- Die Abhaltung einer Lehrveranstaltung als „Blocklehrveranstaltung“ ist nach Genehmigung durch den_die Studiendekan_in möglich. [§ 9 SSB (Module und Lehrveranstaltungen)]
- Die Abhaltung von Lehrveranstaltungen und Prüfungen in einer Fremdsprache ist nach Genehmigung durch den_die Studiendekan_in möglich. [§ 11 SSB (Fremdsprachen)]
- Lehrveranstaltungsprüfungen dienen dem Nachweis der Lernergebnisse, die durch eine einzelne Lehrveranstaltung vermittelt wurden. [§ 12 SSB (Lehrveranstaltungsprüfung)]
- Die Lehrveranstaltungsprüfungen sind von dem_der Leiter_in der Lehrveranstaltung abzuhalten. Bei Bedarf hat das Studienrechtliche Organ eine_n andere_n fachlich geeignete_n Prüfer_in zu bestellen. [§ 12 SSB (Lehrveranstaltungsprüfung)]
- Jedenfalls sind für Prüfungen in Pflicht- und Wahlpflichtlehrveranstaltungen, die in einem einzigen Prüfungsakt enden, drei Prüfungstermine für den Anfang, für die Mitte und für das Ende jedes Semester anzusetzen. Diese sind mit Datum vor Semesterbeginn bekannt zu geben. [§ 15 SSB (Prüfungstermine)]

- Prüfungen dürfen auch am Beginn und am Ende lehrveranstaltungsfreier Zeiten abgehalten werden. [§ 15 SSB (Prüfungstermine)]
- Die Prüfungstermine sind in geeigneter Weise bekannt zu machen. [§ 15 SSB (Prüfungstermine)]

Beschreibung der Lehrveranstaltungstypen

VO: Vorlesungen sind Lehrveranstaltungen, in denen die Inhalte und Methoden eines Faches unter besonderer Berücksichtigung seiner spezifischen Fragestellungen, Begriffsbildungen und Lösungsansätze vorgetragen werden. Die Prüfung wird mit einem einzigen Prüfungsvorgang durchgeführt. In der Modulbeschreibung ist der Prüfungsvorgang je Lehrveranstaltung (schriftlich oder mündlich, oder schriftlich und mündlich) festzulegen. Bei Vorlesungen herrscht keine Anwesenheitspflicht, das Erreichen der Lernergebnisse muss dennoch gesichert sein.

EX: Exkursionen sind Lehrveranstaltungen, die außerhalb der Räumlichkeiten der TU Wien stattfinden. Sie dienen der Vertiefung von Lehrinhalten im jeweiligen lokalen Kontext.

LU: Laborübungen sind Lehrveranstaltungen, in denen Studierende einzeln oder in Gruppen unter Anleitung von Betreuer_innen experimentelle Aufgaben lösen, um den Umgang mit Geräten und Materialien sowie die experimentelle Methodik des Faches zu lernen. Die experimentellen Einrichtungen und Arbeitsplätze werden zur Verfügung gestellt.

PR: Projekte sind Lehrveranstaltungen, in denen das Verständnis von Teilgebieten eines Faches durch die Lösung von konkreten experimentellen, numerischen, theoretischen oder künstlerischen Aufgaben vertieft und ergänzt wird. Projekte orientieren sich am Qualifikationsprofil des Studiums und ergänzen die Berufsvorbildung bzw. wissenschaftliche Ausbildung.

SE: Seminare sind Lehrveranstaltungen, bei denen sich Studierende mit einem gestellten Thema oder Projekt auseinandersetzen und dieses mit wissenschaftlichen Methoden bearbeiten, wobei eine Reflexion über die Problemlösung sowie ein wissenschaftlicher Diskurs gefordert werden.

UE: Übungen sind Lehrveranstaltungen, in denen konkrete Aufgabenstellungen – beispielsweise rechnerisch, konstruktiv, künstlerisch oder experimentell – zu bearbeiten sind. Dabei werden unter fachlicher Anleitung oder Betreuung die Fähigkeiten und Fertigkeiten der Studierenden zur Anwendung auf konkrete Aufgabenstellungen entwickelt.

VU: Vorlesungen mit integrierter Übung sind Lehrveranstaltungen, in denen die beiden Lehrveranstaltungstypen VO und UE in einer einzigen Lehrveranstaltung kombiniert werden. Der jeweilige Übungs- und Vorlesungsanteil darf ein Viertel des

Umfanges der gesamten Lehrveranstaltungen nicht unterschreiten. Beim Lehrveranstaltungstyp VU ist der Übungsteil jedenfalls prüfungsimmanent, der Vorlesungsteil kann in einem Prüfungsakt oder prüfungsimmanent geprüft werden. Unzulässig ist es daher, den Übungsteil und den Vorlesungsteil gemeinsam in einem einzigen Prüfungsvorgang zu prüfen.

Beschreibung der Lehrveranstaltungen und Prüfungen im Informationssystem zu Studien und Lehre

- Typ der Lehrveranstaltung (VO, EX, LU, PR, SE, UE, VU)
- Form (Präsenz, Online, Hybrid, Blended)
- Termine (gegebenenfalls auch die für die positive Absolvierung erforderliche Anwesenheit)
- Inhalte (Beschreibung der Inhalte, Vorkenntnisse)
- Literaturangaben
- Lernergebnisse (Umfassende Beschreibung der Lernergebnisse)
- Methoden (Beschreibung der Methoden in Abstimmung mit Lernergebnissen und Leistungsnachweis)
- Leistungsnachweis (in Abstimmung mit Lernergebnissen und Methoden)
  - Ausweis der Teilleistungen, inklusive Kennzeichnung, welche Teilleistungen wiederholbar sind. Bei Typ VO entfällt dieser Punkt.
- Prüfungen:
  - Inhalte (Beschreibung der Inhalte, Literaturangaben)
  - Form (Präsenz, Online)
  - Prüfungsart bzw. Modus
    - Typ VO: schriftlich, mündlich oder schriftlich und mündlich;
    - bei allen anderen Typen: Ausweis der Teilleistungen inklusive Art und Modus bezugnehmend auf die in der Lehrveranstaltung angestrebten Lernergebnisse.
  - Termine
  - Beurteilungskriterien und Beurteilungsmaßstäbe

§ 7 Prüfungsordnung

Der positive Abschluss des Masterstudiums erfordert:

1. die positive Absolvierung der im Studienplan vorgeschriebenen Module, wobei ein Modul als positiv absolviert gilt, wenn die ihm gemäß Modulbeschreibung zuzurechnenden Lehrveranstaltungen positiv absolviert wurden, sowie die positive Absolvierung der Lehrveranstaltung *Seminar für Diplomand_innen*,

2. die Abfassung einer positiv beurteilten Diplomarbeit und

3. die positive Absolvierung der kommissionellen Abschlussprüfung. Diese erfolgt mündlich vor einem Prüfungssenat gemäß § 13 und § 19 der *Studienrechtlichen Bestimmungen der Satzung der Technischen Universität Wien* und dient der Präsentation und Verteidigung der Diplomarbeit und dem Nachweis der Beherrschung des wissenschaftlichen Umfeldes. Dabei ist vor allem auf Verständnis und Überblickswissen Bedacht zu nehmen. Die Anmeldevoraussetzungen zur kommissionellen Abschlussprüfung gemäß § 17 (1) der *Studienrechtlichen Bestimmungen der Satzung der Technischen Universität Wien* sind erfüllt, wenn die Punkte 1 und 2 erbracht sind.

Das Abschlusszeugnis beinhaltet

(a) die Prüfungsfächer mit ihrem jeweiligen Umfang in ECTS-Punkten und ihren Noten,

(b) das Thema und die Note der Diplomarbeit,

(c) die Note der kommissionellen Abschlussprüfung,

(d) die Gesamtbeurteilung sowie

(e) auf Antrag des_der Studierenden die Gesamtnote des absolvierten Studiums gemäß §72a UG.

Die Note des Prüfungsfaches „Diplomarbeit“ ergibt sich aus der Note der Diplomarbeit. Die Note jedes anderen Prüfungsfaches ergibt sich durch Mittelung der Noten jener Lehrveranstaltungen, die dem Prüfungsfach über die darin enthaltenen Module zuzuordnen sind, wobei die Noten mit dem ECTS-Umfang der Lehrveranstaltungen gewichtet werden. Bei einem Nachkommateil kleiner gleich 0,5 wird abgerundet, andernfalls wird aufgerundet. Wenn keines der Prüfungsfächer schlechter als mit „gut“ und mindestens die Hälfte mit „sehr gut“ benotet wurde, so lautet die *Gesamtbeurteilung* „mit Auszeichnung bestanden“ und ansonsten „bestanden“.

Lehrveranstaltungen des Typs VO (Vorlesung) werden aufgrund einer abschließenden mündlichen und/oder schriftlichen Prüfung beurteilt. Alle anderen Lehrveranstaltungen besitzen immanenten Prüfungscharakter, d.h., die Beurteilung erfolgt laufend durch eine begleitende Erfolgskontrolle sowie optional durch eine zusätzliche abschließende Teilprüfung.

Der positive Erfolg von Prüfungen und wissenschaftlichen sowie künstlerischen Arbeiten ist mit „sehr gut“ (1), „gut“ (2), „befriedigend“ (3) oder „genügend“ (4), der negative Erfolg ist mit „nicht genügend“ (5) zu beurteilen. Bei Lehrveranstaltungen, bei denen eine Beurteilung in der oben genannten Form nicht möglich ist, werden diese durch „mit Erfolg teilgenommen“ (E) bzw. „ohne Erfolg teilgenommen“ (O) beurteilt.

§ 8 Studierbarkeit und Mobilität

Studierende des Masterstudiums *Software Engineering* sollen ihr Studium mit angemessenem Aufwand in der dafür vorgesehenen Zeit abschließen können.

Den Studierenden wird empfohlen, ihr Studium nach dem Semestervorschlag in Anhang C zu absolvieren.

Die Beurteilungs- und Anwesenheitsmodalitäten von Lehrveranstaltungen der Typen UE, LU, PR, VU, SE und EX werden im Rahmen der Lehrvereinbarungen mit dem Studienrechtlichen Organ festgelegt und im Informationssystem für Studien und Lehre bekanntgegeben. Bezüglich der Wiederholbarkeit von Teilleistungen wird auf die studienrechtlichen Bestimmungen der Satzung verwiesen.

Die Anerkennung von im Ausland absolvierten Studienleistungen erfolgt durch das Studienrechtliche Organ. Zur Erleichterung der Mobilität stehen die in § 27 Abs. 1 bis 3 der *Studienrechtlichen Bestimmungen* der Satzung der Technischen Universität Wien angeführten Möglichkeiten zur Verfügung. Diese Bestimmungen können in Einzelfällen auch zur Verbesserung der Studierbarkeit eingesetzt werden.

Die im Zuge einer Mobilität erreichten ECTS-Punkte können verwendet werden, um die im Modul „Freie Wahlfächer und Transferable Skills“ geforderten Transferable Skills im entsprechenden Ausmaß abzudecken. Insbesondere können sie auch dem Themenpool Technikfolgenabschätzung, Technikgenese, Wissenschaftsethik, Gender Mainstreaming und Diversity Management zugerechnet werden.

Ist in einer Lehrveranstaltung die Beschränkung der Teilnehmer_innenzahl erforderlich und kann diese zu Studienzeitverzögerungen führen, sind entsprechend UG § 58 Abs. 8 die Anzahl der Plätze und die Vergabemodalitäten im Studienplan in der jeweiligen Modulbeschreibung vermerkt.

Kommt es in einer Lehrveranstaltung ohne explizit geregelte Platzvergabe zu einem unvorhergesehenen Andrang, kann die Lehrveranstaltungsleitung in Absprache mit dem studienrechtlichen Organ Teilnahmebeschränkungen vornehmen. Studierende, die zum Masterstudium *Software Engineering* zugelassen sind und für die eine Nichtteilnahme zu einer Studienzeitverzögerung führen könnte, werden in jedem Fall aufgenommen. Die Vergabe der allenfalls übrigen Plätze ist nach folgenden Kriterien (mit absteigender Priorität) zu regeln.

- Es werden jene Studierenden bevorzugt aufgenommen, die die formalen und inhaltlichen Voraussetzungen erfüllen. Die inhaltlichen Voraussetzungen können etwa an Hand von bereits abgelegten Prüfungen oder durch einen Eingangstest überprüft werden.
- Unter diesen hat die Verwendung der Lehrveranstaltung als Pflichtfach Vorrang vor der Verwendung als Wahlfach und diese vor der Verwendung als Freifach.
- Innerhalb dieser drei Gruppen sind jeweils jene Studierenden zu bevorzugen, die trotz Vorliegens aller Voraussetzungen bereits in einem früheren Abhaltesemester abgewiesen wurden.

Die Studierenden sind darüber ehebaldigst zu informieren.

§ 9 Diplomarbeit

Die Diplomarbeit ist eine künstlerisch-wissenschaftliche Arbeit, die dem Nachweis der Befähigung dient, ein Thema selbstständig inhaltlich und methodisch vertretbar zu bearbeiten. Das Thema der Diplomarbeit ist von der oder dem Studierenden frei wählbar und muss im Einklang mit dem Qualifikationsprofil stehen.

Das Prüfungsfach *Diplomarbeit* umfasst 30 ECTS-Punkte und besteht aus der wissenschaftlichen Arbeit (Diplomarbeit), die mit 27 ECTS-Punkten bewertet wird, aus der kommissionellen Abschlussprüfung im Ausmaß von 1,5 ECTS-Punkten und einem „Seminar für Diplomand_innen“ im Ausmaß von 1,5 ECTS-Punkten.

§ 10 Akademischer Grad

Den Absolvent_innen des Masterstudiums *Software Engineering* wird der akademische Grad „Diplom-Ingenieur“/„Diplom-Ingenieurin“ – abgekürzt „Dipl.-Ing.“ oder „DI“ (international vergleichbar mit „Master of Science“) – verliehen.

§ 11 Qualitätsmanagement

Das Qualitätsmanagement des Masterstudiums *Software Engineering* gewährleistet, dass das Studium in Bezug auf die studienbezogenen Qualitätsziele der TU Wien konsistent konzipiert ist und effizient und effektiv abgewickelt sowie regelmäßig überprüft wird. Das Qualitätsmanagement des Studiums erfolgt entsprechend dem Plan-Do-Check-Act Modell nach standardisierten Prozessen und ist zielgruppenorientiert gestaltet. Die Zielgruppen des Qualitätsmanagements sind universitätsintern die Studierenden und die Lehrenden sowie extern die Gesellschaft, die Wirtschaft und die Verwaltung, einschließlich des Arbeitsmarktes für die Studienabgänger_innen.

In Anbetracht der definierten Zielgruppen werden sechs Ziele für die Qualität der Studien an der Technischen Universität Wien festgelegt: (1) In Hinblick auf die Qualität und Aktualität des Studienplans ist die Relevanz des Qualifikationsprofils für die Gesellschaft und den Arbeitsmarkt gewährleistet. In Hinblick auf die Qualität der inhaltlichen Umsetzung des Studienplans sind (2) die Lernergebnisse in den Modulen des Studienplans geeignet gestaltet, um das Qualifikationsprofil umzusetzen, (3) die Lernaktivitäten und -methoden geeignet gewählt, um die Lernergebnisse zu erreichen, und (4) die Leistungsnachweise geeignet, um die Erreichung der Lernergebnisse zu überprüfen. (5) In Hinblick auf die Studierbarkeit der Studienpläne sind die Rahmenbedingungen gegeben, um diese zu gewährleisten. (6) In Hinblick auf die Lehrbarkeit verfügt das Lehrpersonal über fachliche und zeitliche Ressourcen um qualitätsvolle Lehre zu gewährleisten.

Um die Qualität der Studien zu gewährleisten, werden der Fortschritt bei Planung, Entwicklung und Sicherung aller sechs Qualitätsziele getrennt erhoben und publiziert. Die Qualitätssicherung überprüft die Erreichung der sechs Qualitätsziele. Zur Messung des ersten und zweiten Qualitätszieles wird von der Studienkommission zumindest einmal pro

Funktionsperiode eine Überprüfung des Qualifikationsprofils und der Modulbeschreibungen vorgenommen. Zur Überprüfung der Qualitätsziele zwei bis fünf liefert die laufende Bewertung durch Studierende, ebenso wie individuelle Rückmeldungen zum Studienbetrieb an das Studienrechtliche Organ, laufend ein Gesamtbild über die Abwicklung des Studienplans. Die laufende Überprüfung dient auch der Identifikation kritischer Lehrveranstaltungen, für welche in Abstimmung zwischen Studienrechtlichem Organ, Studienkommission und Lehrveranstaltungsleiter_innen geeignete Anpassungsmaßnahmen abgeleitet und umgesetzt werden. Das sechste Qualitätsziel wird durch qualitätssichernde Instrumente im Personalbereich abgedeckt. Zusätzlich zur internen Qualitätssicherung wird alle sieben Jahre eine externe Evaluierung der Studien vorgenommen.

Jedes Modul besitzt eine_n Modulverantwortliche_n. Diese Person ist für die inhaltliche Kohärenz und die Qualität der dem Modul zugeordneten Lehrveranstaltungen verantwortlich. Diese wird insbesondere durch zyklische Kontrollen, inhaltliche Feinabstimmung mit vorausgehenden und nachfolgenden Modulen sowie durch Vergleich mit analogen Lehrveranstaltungen bzw. Modulen anderer Universitäten im In- und Ausland sichergestellt.

Lehrveranstaltungskapazitäten

Um die Qualität der Umsetzung der Lehrveranstaltungen zu sichern, dienen für die verschiedenen Typen von Lehrveranstaltungen (siehe Seite 27) die folgenden Gruppengrößen als Richtwert:

	Gruppengröße	
Lehrveranstaltungstyp	je Leiter(in)	je Tutor(in)
VO	100	
UE mit Tutor(inn)en	30	15
UE	15	
LU mit Tutor(inn)en	20	8
LU	8	
EX, PR, SE	10	

Für Lehrveranstaltungen des Typs VU werden für den Vorlesungs- bzw. Übungsteil die Gruppengrößen für VO bzw. UE herangezogen. Die Beauftragung der Lehrenden erfolgt entsprechend der tatsächlichen Abhaltung.

Gender, Ethik und Diversität

Kontext: Um Lehr- und Lernumgebungen zu schaffen, in denen alle Studierenden – unabhängig von Geschlecht, Herkunft, Fähigkeiten oder sozialem Hintergrund – gleichermaßen geschätzt, gefördert und gefordert werden, ist eine inklusive Lehre basierend auf

diversitätssensibler Didaktik erforderlich. Dies kann nicht in eigenen separaten Lehrveranstaltungen abgehandelt werden, sondern muss auf allen Ebenen des Studiums umgesetzt werden – als „Embedded Gender, Ethics and Diversity“.

Dazu gehört die Einbettung ethischer und gesellschaftlicher Themen in den Studienplan, sowie die Auseinandersetzung mit diesen Themen in jeder Lehrveranstaltung. So lassen sich Rahmenbedingungen schaffen, die ein diskriminierungsfreies Lernumfeld ermöglichen. Dies umfasst auch Maßnahmen gegen Diskriminierung und Belästigung, etwa durch explizite Verhaltenscodizes.

Lehrinhalt: Inklusivität und Vielfalt werden gefördert, indem in allen Lehrveranstaltungen unterschiedliche Perspektiven einbezogen werden, die sich auf ein breites Spektrum von Autor_innen und Rollenvorbildern stützen. Die ethische Reflexion von Kernbereichen wird in allen Lehrveranstaltungen eingebettet, indem sie in Vorlesungen und Übungsbeispielen angesprochen und berücksichtigt wird, etwa durch Diskussion ethischer Aspekte und sozialer Auswirkungen. Zusätzlich wird auf das Angebot der Abteilung für Genderkompetenz der TU Wien hingewiesen, die für das Absolvieren eines Zusatzkatalogs das Zertifikat *Gender- und Diversitätskompetenz* ausstellt.

Lehrmethoden: Die Lehrmethoden fördern Gender-Inklusivität und Diversität und schaffen ein Lernumfeld, in dem sich alle Studierenden gleichermaßen willkommen fühlen. Dazu gehören unter anderem die Verwendung einer inklusiven, vorurteilsfreien Sprache, um die Verstärkung von Stereotypen zu vermeiden; Lehrmaterial mit Beispielen, Fallstudien oder Anschauungsmaterial, die unsere vielfältige Gesellschaft widerspiegeln; die Abhaltung der Lehre in einer Form, die für Menschen mit unterschiedlichen Fähigkeiten geeignet ist und eine gleichberechtigte Teilnahme gewährleistet; sowie die Einbindung von Gastredner_innen mit unterschiedlichen Hintergründen.

§ 12 Inkrafttreten

Dieser Studienplan tritt mit 1. Oktober 2025 in Kraft.

§ 13 Übergangsbestimmungen

Die Übergangsbestimmungen sind in Anhang B zu finden.

A Modulbeschreibungen

Die den Modulen zugeordneten Lehrveranstaltungen werden in folgender Form angeführt:
9,9/9,9 XX Titel der Lehrveranstaltung
Dabei bezeichnet die erste Zahl den Umfang der Lehrveranstaltung in ECTS-Punkten und die zweite ihren Umfang in Semesterstunden. ECTS-Punkte sind ein Maß für den Arbeitsaufwand der Studierenden, wobei ein Studienjahr 60 ECTS-Punkte umfasst und ein ECTS-Punkt 25 Stunden zu je 60 Minuten entspricht. Eine Semesterstunde entspricht so vielen Unterrichtseinheiten wie das Semester Unterrichtswochen umfasst. Eine Unterrichtseinheit dauert 45 Minuten. Der Typ der Lehrveranstaltung (XX) ist in § 6 unter *Lehrveranstaltungstypen* auf Seite 27 im Detail erläutert.

Advanced Cryptography

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module expands the students' basic knowledge and skills in the area of cryptography.

Fachkompetenzen: Upon successful completion of this module, students are able to describe topics and concepts (see below) in cryptography, which form the basis for interacting in a privacy-preserving way in the digital world; they are capable of evaluating systems and analyzing their security by applying the provable-security framework and its extensions.

Überfachliche Kompetenzen: Students learn to argue in a rigorous way, by first modeling desired notions and then giving formal guarantees (e.g. by using the complexity-theoretic notion of "proofs by reduction") against adversaries whose attack strategies are unknown to the system designer. Furthermore, students are able to identify and discuss ethical questions in the context of cryptography.

Inhalt:

- Provable security, the random-oracle model;
- elliptic-curve-based cryptography;
- zero-knowledge and succinct proof systems;
- secure multi-party computation;
- post-quantum (lattice-based) cryptography.

Erwartete Vorkenntnisse: Knowledge of the basic concepts of cryptography in symmetric and public-key encryption and authentication as well as familiarity with the concept of provable security.

These prerequisites are taught in the module *Introduction to Cryptography*

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Advanced Cryptography

Advanced Database Systems

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with different database system technologies - including classical relational database systems, NoSQL systems, and systems specifically designed for distributed data processing in a cluster.

Fachkompetenzen: After successful completion of the module, students

- can use advanced features and extensions of relational data management such as recursion, stored procedures and triggers, object-orientation and graphs,
- know central "big data" concepts, methods, technologies, and subject-specific terminology,
- are able to use distributed data processing methods such as MapReduce and Spark,
- can name basic principles of NoSQL systems,
- are able to use various NoSQL technologies.

Überfachliche Kompetenzen: Students acquire the ability to understand different paradigms underlying database system technologies and choosing and managing an appropriate one for a given application. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt: The course covers three major topics:

- Advanced features and extensions of relational data management,
- Introduction to distributed data processing techniques (with a focus on MapReduce and Spark),
- Basic principles of various NoSQL systems.

Erwartete Vorkenntnisse: Basic knowledge of "Database Systems", "Algorithms and Data Structures", and "Programming"

These prerequisites are taught in the following modules:

- Database Systems (*Datenbanksysteme*),
- Algorithms and Data Structures (*Algorithmen und Datenstrukturen*),
- Programming (*Einführung in die Programmierung*).

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Advanced Database Systems

Advanced Human-Centered AI: from concepts to implementation

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with human-centred AI as well as applications thereof.

Fachkompetenzen: After successful completion of the course, students are able to:

- Explain user interface basics and AI system design principles.
- Describe fundamentals of supervised, unsupervised, and reinforcement learning.
- Explain and apply gesture recognition (sequence classification, Markov property, gesture and pose detection from video).
- Implement adaptive user interfaces (automated optimization based on human factor parameters).
- Work with tools and frameworks (TensorFlow, Unity ML-Agents).

Überfachliche Kompetenzen:

- Collaborate effectively in interdisciplinary teams to address complex design problems.
- Designing AI systems that effectively support or cooperate with human users.
- Communicate design concepts and processes clearly and persuasively to diverse audiences.
- Adapt to feedback and iterative processes with resilience and a growth mindset.

Inhalt:

- Basics of AI systems: supervised, unsupervised, and reinforcement learning.
- Recommender systems: definitions, collaborative filtering, similarity measures.
- Natural language processing: syntax, semantics, tokenization, normalization, stemming, chatbot interaction.
- Gesture recognition: sequence classification, Markov property, gesture and pose recognition from video.
- Adaptive user interfaces: automated optimization based on human factor parameters.
- Explainable AI: local and global interpretability, LIME, SHAP, automated rationale generation.

Erwartete Vorkenntnisse: Knowledge of “Software Engineering” and “Programming”

These prerequisites are taught in the following modules:

- Software Engineering (*Software Engineering*)
- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Advanced Human-Centered AI: from concepts to implementation

Advanced Internet Computing

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: Advanced Internet Computing deals with theoretical foundations, technologies, architectures, standards and use cases of contemporary and next-generation distributed Internet computing systems that span the computing continuum. After successful completion of the module, students are able to utilize and be knowledgeable in Service-oriented Computing, Web services, Cloud Computing, the IoT-Cloud spectrum as well as advanced topics.

Fachkompetenzen: Designing, building and operating intelligent Internet computing services and managing the compute/communication infrastructure that supports them.

Überfachliche Kompetenzen: Students build skills across various aspects of contemporary Internet computing systems, particularly on computation, communication, and service design/implementation/orchestration. These skills can be applied to diverse application domains within the IoT. Via lectures and group projects, students further learn to deal with the intersection between AI/ML and edge/cloud computing. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Service-oriented Computing and Service-oriented Architecture.
- Cloud, edge and fog computing.
- Service design and delivery paradigms: From microservices to serverless computing.
- Internet of Things: Architectures, supporting technologies and applications.
- IoT-Cloud Continuum: Computation, communication, service design and management aspects.
- Edge computing and AI/ML at the edge: Edge-tailored ML model training and serving mechanisms; using AI/ML to improve operations in the computing continuum.

Erwartete Vorkenntnisse: Significant knowledge in programming (e.g., Java / .NET/ Python) as well as distributed systems are required.

- Distributed Systems (*Verteilte Systeme*)
- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Advanced Internet Computing

Advanced Logic Programming

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with advanced programming techniques in ISO Prolog.

Fachkompetenzen: Develop higher-order Prolog programs, including lambda expressions. Apply meta programming techniques starting from meta-circular interpreters. Classify and apply program transformation techniques.

Überfachliche Kompetenzen: Students acquire the ability to use advanced programming techniques and apply them for new domains. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- All solutions predicates
- Meta programming
- Higher-order programming
- Lambda expressions
- Reification
- Meta interpreters
- Program transformations

Erwartete Vorkenntnisse: Knowledge of "Logic programming and constraints"

These prerequisites are taught in the following module:

- Logic programming and constraints (*Logikprogrammierung und Constraints*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Advanced Logic Programming

Advanced Model Engineering

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with advanced model-driven approaches to software engineering. It combines techniques, methods, and tools from language engineering, model engineering, web engineering, and AI.

Fachkompetenzen: Advanced concepts and techniques of model-driven software engineering and the application thereof.

After successful completion of the module, students are able to

- describe the taught advanced topics of model engineering (like ontology-driven conceptual modeling and multi-level modeling),
- describe the taught concepts and techniques for web modeling (like the Language Server Protocol (LSP), Langium, and the Graphical Language Server Platform (GLSP), and
- apply their gained knowledge to realize a web-based modeling environment.

Überfachliche Kompetenzen: Students acquire the ability to explain methods for advanced model-driven software engineering and design domain-specific, AI-assisted model-driven software engineering solutions through web technologies. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Ontology-driven conceptual modeling
- Multi-level modeling
- Langium
- Language Server Protocol (LSP) and Graphical Language Server Platform (GLSP)
- Web modeling
- Web Model Visualization
- Web Model Interaction
- Development of plugins for LSP-based ecosystems like Theia and VS Code

Erwartete Vorkenntnisse: Knowledge of "Software Engineering", "Object-oriented modeling", and "Model engineering"

These prerequisites are taught in the following modules:

- Software Engineering (*Software Engineering*)
- Programming (*Einführung in die Programmierung*)
- Model Engineering (*Model Engineering*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Advanced Model Engineering

Advanced Multiprocessor Programming

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: The module aims to provide a deeper understanding of behavior of and ways to reason about shared-memory multi-core processors, as well to supply tools to implement efficient and correct algorithms for such processors. Classical, loosely synchronous parallel paradigms are contrasted with methods for providing lock- and wait-free guarantees. Practical experience with lock-free algorithms and data structures will be gained through practical programming projects using standard multi-core processor systems.

Fachkompetenzen: Mutual exclusion and similar fundamental problems will be introduced to reason about the synchronization power of registers and various atomic operations, centered around the consensus problem. The insights are used to study a selection of standard lock- and wait-free algorithms and data structures that can be of practical relevance. Specialized data structures for supporting paradigms like work-stealing may be introduced. Memory behavior, synchronization, and the memory reclamation problem will be covered. Thread-based programming frameworks like OpenMP will be used for the practical parts.

Überfachliche Kompetenzen: Students will gain a deeper understanding of the intricacy of modern,shared-memory multi-processor systems, and gain a deeper appreciation of the sometimes conflicting goals of concurrent and parallel computing.

Inhalt: After successful completion of the module, students are able to

- Appreciate fundamental synchronization and coordination problems for shared-memory multiprocessors, including fundamental limitations
- Understand, exploit and cope with memory models and behavior
- Understand and apply basic concepts for lock-based, lock- and wait-free algorithms
- Apply fundamental lock-free algorithms and data structures (lists, stacks, queues...)
- Understand concepts and implementations of work-stealing schedulers

Erwartete Vorkenntnisse: Basic knowledge in parallel computing, computer architecture, software engineering, programming (in C or related languages) and algorithms.

These prerequisites are taught in the following modules:

- Parallel Computing (*Parallel Computing*)
- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Advanced Multiprocessor Programming

Advanced Topics In Algorithms and Complexity

Regelarbeitsaufwand: min. 3,0 ECTS

Lernergebnisse: The module provides in-depth knowledge and understanding of important sub-areas of algorithms and complexity.

Fachkompetenzen:

- Students can name, explain and contrast the most important theories, principles, concepts and algorithms and complexity in specific topics of algorithms and complexity. Their knowledge and understanding corresponds to the state of the art in algorithms and complexity literature.

In particular, students can

- collect, develop, evaluate and interpret information relevant to algorithms and complexity,
- determine requirements and constraints in different areas of algorithms and complexity, and
- practically apply the acquired knowledge in complex algorithms and complexity tasks, work out and further develop solutions to problems and argue for these solutions.

Überfachliche Kompetenzen:

- Students are able to deepen their knowledge independently.

Inhalt: This module deepens knowledge in selected areas of algorithms and complexity. The exact selection of topics varies from term to term.

Erwartete Vorkenntnisse: The expected competencies and knowledge depend on the chosen courses.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: The courses for this module are announced before each term.

Advanced Topics In Automation and Mobile Robotics

Regelarbeitsaufwand: min. 3,0 ECTS

Lernergebnisse: The module provides in-depth knowledge and understanding of important sub-areas of automation and mobile robotics.

Fachkompetenzen:

- Students can name, explain and contrast the most important theories, principles, concepts and automation and mobile robotics in specific topics of automation and mobile robotics. Their knowledge and understanding corresponds to the state of the art in automation and mobile robotics literature.

In particular, students can

- collect, develop, evaluate and interpret information relevant to automation and mobile robotics,
- determine requirements and constraints in different areas of automation and mobile robotics, and
- practically apply the acquired knowledge in complex automation and mobile robotics tasks, work out and further develop solutions to problems and argue for these solutions.

Überfachliche Kompetenzen:

- Students are able to deepen their knowledge independently.

Inhalt: This module deepens knowledge in selected areas of automation and mobile robotics. The exact selection of topics varies from term to term.

Erwartete Vorkenntnisse: The expected competencies and knowledge depend on the chosen courses.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: The courses for this module are announced before each term.

Advanced Topics In Data Management and Intelligent Systems

Regelarbeitsaufwand: min. 3,0 ECTS

Lernergebnisse: The module provides in-depth knowledge and understanding of important sub-areas of data management and intelligent systems.

Fachkompetenzen:

- Students can name, explain and contrast the most important theories, principles, concepts and data management and intelligent systems in specific topics of data management and intelligent systems. Their knowledge and understanding corresponds to the state of the art in data management and intelligent systems literature.

In particular, students can

- collect, develop, evaluate and interpret information relevant to data management and intelligent systems,
- determine requirements and constraints in different areas of data management and intelligent systems, and
- practically apply the acquired knowledge in complex data management and intelligent systems tasks, work out and further develop solutions to problems and argue for these solutions.

Überfachliche Kompetenzen:

- Students are able to deepen their knowledge independently.

Inhalt: This module deepens knowledge in selected areas of data management and intelligent systems. The exact selection of topics varies from term to term.

Erwartete Vorkenntnisse: The expected competencies and knowledge depend on the chosen courses.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: The courses for this module are announced before each term.

Advanced Topics In Distributed and Next Generation Computing

Regelarbeitsaufwand: min. 3,0 ECTS

Lernergebnisse: The module provides in-depth knowledge and understanding of important sub-areas of distributed and next generation computing.

Fachkompetenzen:

- Students can name, explain and contrast the most important theories, principles, concepts and distributed and next generation computing in specific topics of distributed and next generation computing. Their knowledge and understanding corresponds to the state of the art in distributed and next generation computing literature.

In particular, students can

- collect, develop, evaluate and interpret information relevant to distributed and next generation computing,
- determine requirements and constraints in different areas of distributed and next generation computing, and
- practically apply the acquired knowledge in complex distributed and next generation computing tasks, work out and further develop solutions to problems and argue for these solutions.

Überfachliche Kompetenzen:

- Students are able to deepen their knowledge independently.

Inhalt: This module deepens knowledge in selected areas of distributed and next generation computing. The exact selection of topics varies from term to term.

Erwartete Vorkenntnisse: The expected competencies and knowledge depend on the chosen courses.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: The courses for this module are announced before each term.

Advanced Topics In High Performance Computing

Regelarbeitsaufwand: min. 3,0 ECTS

Lernergebnisse: The module provides in-depth knowledge and understanding of important sub-areas of high performance computing.

Fachkompetenzen:

- Students can name, explain and contrast the most important theories, principles, concepts and high performance computing in specific topics of high performance computing. Their knowledge and understanding corresponds to the state of the art in high performance computing literature.

In particular, students can

- collect, develop, evaluate and interpret information relevant to high performance computing,
- determine requirements and constraints in different areas of high performance computing, and
- practically apply the acquired knowledge in complex high performance computing tasks, work out and further develop solutions to problems and argue for these solutions.

Überfachliche Kompetenzen:

- Students are able to deepen their knowledge independently.

Inhalt: This module deepens knowledge in selected areas of high performance computing. The exact selection of topics varies from term to term.

Erwartete Vorkenntnisse: The expected competencies and knowledge depend on the chosen courses.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: The courses for this module are announced before each term.

Advanced Topics In Machine Learning

Regelarbeitsaufwand: min. 3,0 ECTS

Lernergebnisse: The module provides in-depth knowledge and understanding of important sub-areas of machine learning.
Fachkompetenzen:

- Students can name, explain and contrast the most important theories, principles, concepts and algorithms in specific topics of machine learning. Their knowledge and understanding corresponds to the state of the art in machine learning literature.

In particular, students can

- collect, develop, evaluate and interpret information relevant to machine learning,
- determine requirements and constraints in different areas of machine learning, and
- practically apply the acquired knowledge in complex machine learning tasks, work out and further develop solutions to problems and argue for these solutions.

Überfachliche Kompetenzen:

- Students are able to deepen their knowledge independently.

Inhalt: This module deepens knowledge in selected areas of machine learning. The exact selection of topics varies from term to term.

Erwartete Vorkenntnisse: The expected competencies and knowledge depend on the chosen courses.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: The courses for this module are announced before each term.

Advanced Topics In Security and Privacy

Regelarbeitsaufwand: min. 3,0 ECTS

Lernergebnisse: The module provides in-depth knowledge and understanding of important sub-areas of security and privacy.

Fachkompetenzen:

- Students can name, explain and contrast the most important theories, principles, concepts and security and privacy in specific topics of security and privacy. Their knowledge and understanding corresponds to the state of the art in security and privacy literature.

In particular, students can

- collect, develop, evaluate and interpret information relevant to security and privacy,
- determine requirements and constraints in different areas of security and privacy, and
- practically apply the acquired knowledge in complex security and privacy tasks, work out and further develop solutions to problems and argue for these solutions.

Überfachliche Kompetenzen:

- Students are able to deepen their knowledge independently.

Inhalt: This module deepens knowledge in selected areas of security and privacy. The exact selection of topics varies from term to term.

Erwartete Vorkenntnisse: The expected competencies and knowledge depend on the chosen courses.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: The courses for this module are announced before each term.

Advanced Topics In Societal Impact and Critical Reflections

Regelarbeitsaufwand: min. 3,0 ECTS

Lernergebnisse: The module provides in-depth knowledge and understanding of important sub-areas of societal impact and critical reflections.

Fachkompetenzen:

- Students can name, explain and contrast the most important theories, principles, concepts and algorithms in specific topics of societal impact and critical reflections. Their knowledge and understanding corresponds to the state of the art in societal impact and critical reflections literature.

In particular, students can

- collect, develop, evaluate and interpret information relevant to societal impact and critical reflections,
- determine requirements and constraints in different areas of societal impact and critical reflections, and
- practically apply the acquired knowledge in complex societal impact and critical reflections tasks, work out and further develop solutions to problems and argue for these solutions.

Überfachliche Kompetenzen:

- Students are able to deepen their knowledge independently.

Inhalt: This module deepens knowledge in selected areas of societal impact and critical reflections. The exact selection of topics varies from term to term.

Erwartete Vorkenntnisse: The expected competencies and knowledge depend on the chosen courses.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: The courses for this module are announced before each term.

Advanced Topics In Software Engineering and Programming

Regelarbeitsaufwand: min. 3,0 ECTS

Lernergebnisse: The module provides in-depth knowledge and understanding of important sub-areas of software engineering and programming.

Fachkompetenzen:

- Students can name, explain and contrast the most important theories, principles, concepts and algorithms in specific topics of software engineering and programming. Their knowledge and understanding corresponds to the state of the art in software engineering and programming literature.

In particular, students can

- collect, develop, evaluate and interpret information relevant to software engineering and programming,
- determine requirements and constraints in different areas of software engineering and programming, and
- practically apply the acquired knowledge in complex software engineering and programming tasks, work out and further develop solutions to problems and argue for these solutions.

Überfachliche Kompetenzen:

- Students are able to deepen their knowledge independently.

Inhalt: This module deepens knowledge in selected areas of software engineering and programming. The exact selection of topics varies from term to term.

Erwartete Vorkenntnisse: The expected competencies and knowledge depend on the chosen courses.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: The courses for this module are announced before each term.

Advanced Topics In Verification and Automated Reasoning

Regelarbeitsaufwand: min. 3,0 ECTS

Lernergebnisse: The module provides in-depth knowledge and understanding of important sub-areas of verification and automated reasoning.

Fachkompetenzen:

- Students can name, explain and contrast the most important theories, principles, concepts and algorithms in specific topics of verification and automated reasoning. Their knowledge and understanding corresponds to the state of the art in verification and automated reasoning literature.

In particular, students can

- collect, develop, evaluate and interpret information relevant to verification and automated reasoning,
- determine requirements and constraints in different areas of verification and automated reasoning, and
- practically apply the acquired knowledge in complex verification and automated reasoning tasks, work out and further develop solutions to problems and argue for these solutions.

Überfachliche Kompetenzen:

- Students are able to deepen their knowledge independently.

Inhalt: This module deepens knowledge in selected areas of verification and automated reasoning. The exact selection of topics varies from term to term.

Erwartete Vorkenntnisse: The expected competencies and knowledge depend on the chosen courses.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: The courses for this module are announced before each term.

Advanced Privacy Enhancing Technologies

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: Upon successful completion of this module, students are able to critically analyze privacy risks in data processing systems and apply cryptographic and information-theoretic methods to enhance privacy. They are able to design solutions using techniques such as secret sharing, differential privacy, and secure multiparty computation, and evaluate their effectiveness in practical scenarios. Additionally, they can compare different approaches to privacy-preserving data analysis and, time permitting, implement basic mechanisms for privacy protection in machine learning applications. This knowledge equips students to develop privacy-enhancing technologies that balance data utility with strong privacy guarantees in AI-driven and data-intensive domains.

Fachkompetenzen: Students are able to analyze, design, and evaluate advanced privacy-enhancing technologies such as secret sharing, differential privacy, and secure multiparty computation, and apply them in modern application contexts.

Überfachliche Kompetenzen: Students are able to transfer and adapt methods of privacy-enhancing technologies to new domains, critically assess their applicability, and communicate technical concepts effectively across interdisciplinary teams.

Inhalt:

- Definitions of confidentiality and introduction of the concepts;
- secret sharing;
- differential privacy;
- secure multi-party computation;
- machine learning and privacy.

Erwartete Vorkenntnisse: Students are expected to know the fundamentals of IT security, cryptography, provable security, and privacy-enhancing technologies.

These prerequisites are taught in the bachelor modules

- *Einführung in Security*;
- *Introduction to Cryptography*; and
- *Privacy-Enhancing Technologies*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Advanced Privacy Enhancing Technologies

Advanced Reinforcement Learning

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module provides more advanced insights into reinforcement learning, one of the main fields of machine learning.

Fachkompetenzen: After successful completion of the module, students are able to:

- explain the advanced theory of reinforcement learning;
- implement advanced reinforcement algorithms; and
- apply advanced reinforcment algorithms to realistic problems.

Überfachliche Kompetenzen: Students acquire the ability to understand advanced methods for machine learning and artificial intelligence and to solve complex problems in (stochastic) optimal control. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Distributional reinforcement learning;
- distributional deep reinforcement learning;
- convergence proofs in non-distributional reinforcement learning;
- convergence proofs in distributional reinforcement learning.

Erwartete Vorkenntnisse: Students should already have familiarity with basics of reinforcement learning.

These prerequisites are taught in module *Reinforcement Learning*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Advanced Reinforcement Learning

Advanced Research in Algorithmics

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: Through a combination of theoretical analysis and practical exploration, the course delves into advanced techniques for designing algorithms and analyzing their complexity.

Fachkompetenzen: Students will learn how to

- identify and tackle new and familiar research situations,
- explain and model new computational problems,
- read and understand scientific publications,
- identify relevant research questions for concrete problems in algorithms and complexity,
- carry out independent scientific research on concrete problems in algorithms and complexity,
- cooperate with other team members and with advisors on research,
- present their results in a clear and concise manner, both orally and in writing, and
- classify and assess recent research results as well as their own results.

Überfachliche Kompetenzen: Students will be able to perform

- formal reasoning and critical analysis of algorithmic approaches,
- structured problem-solving in complex, research-oriented contexts,
- abstract modeling of computational and optimization problems,
- evaluation of trade-offs between solution quality, efficiency, and resource usage, and
- have awareness of ethical, fairness, and societal issues in algorithmic decision-making.

Inhalt: The module introduces students to research-oriented work in algorithmics. Students engage with the topics through literature review, critical analysis, the development of preliminary research ideas. Additional technical results and methods useful for tackling the individual topics may be supplied by the advisors and lecturers.

The research topics come from a range of areas such as computational social choice, algorithmic game theory, computational geometry, graph algorithms, and approximation techniques.

Representative topics include:

- voting rules and algorithmic aspects of elections;
- cooperative games and coalition formation in multi-agent systems;
- resource allocation and fair division algorithms;
- complexity and approximation of computationally hard problems such as clustering;
- geometric problems such as convex hulls, proximity graphs, and low-dimensional embeddings; and
- graph drawing problems such as planarization and crossing minimization.

Erwartete Vorkenntnisse: Good knowledge from algorithmics and complexity analysis. These prerequisites are taught in the following modules:

- *Algorithmen und Datenstrukturen*;
- *Effiziente Algorithmen* (recommended);
- *Algorithmics*; and
- *Complexity Theory*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Advanced Research in Algorithmics

Advanced Software Engineering

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with reliability techniques for large, complex software systems as well as modern applications thereof.

Fachkompetenzen: Fundamental concepts and techniques on software reliability and modern applications thereof.

Überfachliche Kompetenzen: Students acquire the ability to explain methods for automated software reliability and design them themselves for new application domains. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Static program-analysis techniques, such as deductive verification, abstract interpretation, bounded model checking, and symbolic execution

- Dynamic program-analysis techniques, such as dynamic symbolic execution, greybox fuzzing, and blackbox fuzzing
- Specification-inference techniques
- Program-synthesis techniques
- Modern applications of reliability techniques to different, popular domains, such as smart contracts and machine-learning models

Erwartete Vorkenntnisse: Knowledge of "Software Engineering" and "Programming"
These prerequisites are taught in the following modules:

- Software Engineering (*Software Engineering*)
- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Advanced Software Engineering

Advanced Software Engineering Project

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with engineering large software systems in a group project.

Fachkompetenzen: Realisation of a medium-sized software project in a team of 4-6 persons with intensive use of appropriate tools and methods of modern software development.

Überfachliche Kompetenzen: After successful completion of the module, students shall be able to

- estimate and quantify the properties of large software systems and projects
- apply development procedures typical for large software systems
- explain and implement architectures of large software systems
- apply approaches to developing software for mobile devices
- evaluate and select tools and frameworks for specific problems
- describe and apply tools and frameworks for large software systems

Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Overview on characteristics of advanced software systems
  - Value-based Software Engineering

  - System size and complexity
  - System dependability, Quality of Service (QoS)
  - Extended Software Lifecycle
- Software Engineering approaches for advanced software systems
  - Architecture styles, such as component-based software engineering
  - Lifecycle management and documentation
  - Software design patterns
- Component-based engineering of complex software systems
- Automation in developing advanced software systems
  - Sourcecode Management
  - Executable specification (e.g., Gherkin/Cucumber)
  - Continuous integration and testing
  - Advanced build management
  - Persistence techniques (e.g., object-relational mapping)
  - Data-driven approaches for software engineering and documentation (e.g., with LLMs)
- Aspects of enterprise architectures
- Overview on selected research and industrial topics
  - Software engineering for mobile devices
  - Software-intensive systems
  - Ecosystems for software engineering tools and frameworks

Erwartete Vorkenntnisse: Knowledge of "Software Engineering" and "Programming"

- Software Engineering (*Software Engineering*)
- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 PR Advanced Software Engineering

AI Ethics

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with the various ethical challenges and topics arising in the context of AI. Besides an in-depth introduction to ethical frameworks and AI ethics themes, the module focuses on the development of critical reflection and argumentation/discussion skills.

Fachkompetenzen: After successful completion of the course, students are able to:

- explain and summarize central concepts and problems in AI ethics (e.g., AI alignment, algorithmic bias);
- describe fundamental ethical theories (e.g., utilitarianism, care ethics) and illustrate their application to AI ethics topics through examples;
- analyze and construct a critical ethical reflection on central technical and philosophical challenges in AI ethics (e.g., fairness, sustainability, AI alignment);
- formulate and defend clearly structured, state of the art informed arguments on key AI ethics questions (e.g., the right to explanation, data privacy), engaging with counterarguments; and
- justify the importance of ethical reflection in AI development, using ethical reflection and/or case studies to support claims.

Überfachliche Kompetenzen: Students will acquire critical thinking skills that are strongly rooted in humanity studies. They will also learn how to study text on policy, law, and regulations concerning AI.

This course will address ethics, gender, and diversity in the following ways: The main purpose of the course is for students to develop critical reflection skills on ethical aspects of AI, this includes obtaining a broad an diverse overview of various perspectives on ethical challenges. It reflects explicitly on questions of gender and diversity in terms of fairness and bias both in ethics and in AI. Furthermore, the literature for the course is aimed at reflecting diverse backgrounds too.

Inhalt:

- Novel ethical challenges in AI such as disinformation, explainability, sustainability, data-privacy, autonomy;
- methods such as embedded ethics, multi-scale ethics, AI ethics audits, AI alignment, human-in-the-loop AI;
- ethical theories including consequentialism, utilitarianism, virtue ethics, deontology, care ethics, and critical methods such as feminist ethics.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU AI Ethics

AI Programming

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module provides students with the necessary knowledge and skills to develop AI systems using modern programming approaches.

Fachkompetenzen: Fundamental concepts and techniques of AI programming, including emerging paradigms and methodologies for the development of (probabilistic) models.

Überfachliche Kompetenzen: Students develop the ability to explain and implement AI-driven models using modern programming techniques. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Fundamental concepts of AI programming
- Programming techniques and paradigms for AI system development
- Underlying techniques and models to enable AI for code
- Representation and inference in AI-driven models
- Applications of emerging paradigms such as probabilistic programming and differentiable programming

Erwartete Vorkenntnisse: Knowledge of Machine Learning and Programming

These prerequisites are taught in the following modules:

- Machine Learning (*Einführung in Machine Learning*)
- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU AI Programming

Algorithmic Encoding Techniques

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with systematic approaches to encode hard computational problems into well-studied target formalisms.

Fachkompetenzen: Students gain comprehensive knowledge of encoding techniques and solving strategies. They learn to develop correct and efficient encodings for complex problems and verify their solutions through formal methods.

Überfachliche Kompetenzen: Students acquire the ability to abstract computational problems into formal representations and analyze solution approaches through mathematical reasoning. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender and diversity in the context of algorithmic encoding techniques and their applications.

Inhalt:

- Encoding strategies for target formalisms, including eager and lazy approaches;
- local consistency and propagation;
- static and dynamic symmetry-breaking techniques for improving solving efficiency;
- correctness certification methods for encodings and solutions;
- incremental solving techniques for dynamic problem instances;
- encodings for parallel solving and portfolio-based approaches.

Erwartete Vorkenntnisse: A solid understanding of algorithms and logic is required. These prerequisites are taught in the following modules:

- *Algorithmen und Datenstrukturen*;
- *Discrete Mathematics*; and
- *Algorithmics* (strongly recommended).

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Algorithmic Encoding Techniques

Algorithmic Geometry

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module introduces fundamental efficient algorithms and data structures for combinatorial geometric problems.

Fachkompetenzen: Students will learn to

- explain fundamental concepts, structures, and problems in algorithmic geometry,
- design and analyze geometric algorithms and data structures,
- model geometric problems and adapt known algorithms and data structures to new problems, and
- investigate unknown geometric problems and develop new algorithmic solutions.

Überfachliche Kompetenzen: Students will be able to

- perform formal reasoning, correctness proofs, and proofs of asymptotic time complexity of algorithms,
- structured analysis of computational problems,
- abstract problem modelling,
- judge trade-offs between time and space complexity, and
- being aware of possible ethical, gender, and diversity issues in the context of geometric data processing.

Inhalt: The module content covers fundamental techniques and algorithms for geometric problems including

- basic geometric algorithms for computing convex hulls, polygon triangulations, and segment intersections,
- data structures for orthogonal range searching, windowing queries, and point location queries,
- Voronoi diagrams and Delaunay triangulations,
- point-line duality,
- distance approximation,
- geometric shortest paths and robot motion planning, and
- sweep-line algorithms and randomized incremental constructions.

Erwartete Vorkenntnisse: Extended knowledge of algorithms and data structures.
These prerequisites are taught in the following recommended modules:

- *Effiziente Algorithmen*
- *Algorithmics*

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Algorithmic Geometry

Algorithmic Social Choice

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: The module provides a comprehensive overview of algorithmic and computational approaches to collective decision-making and social choice problems.
Fachkompetenzen: Students will learn to:

- explain fundamental concepts in voting theory, fair division, and matching problems
- analyze computational properties and complexity results in social choice settings
- design and evaluate algorithms for collective decision-making processes
- model social choice problems formally and propose algorithmic solutions

Überfachliche Kompetenzen: Students develop interdisciplinary skills by tackling problems at the intersection of economics, social choice theory, and computer science. They explore diversity-related dimensions of algorithmic decision-making, analyzing how different voting rules, matching algorithms, and fair division procedures can advantage or disadvantage various social groups. Issues of gender representation and fairness are considered as

specific aspects of diversity in algorithmic decision-making. The module emphasizes the design of algorithms that are not only efficient, but also equitable and sensitive to diverse needs.

Inhalt:

- Aggregating preferences (rank aggregation) and voting
- Preference domain restrictions
- Matching under preferences
- Algorithmic mechanism design
- Cake cutting protocols
- Fair allocation of resources
- Judgment aggregation

Erwartete Vorkenntnisse: Knowledge from algorithmics and complexity analysis.

These prerequisites are taught in the following modules:

- *Algorithmen und Datenstrukturen*
- *Effiziente Algorithmen*
- *Algorithmics*
- *Complexity Theory*

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Algorithmic Social Choice

Algorithmics

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with advanced algorithmic methods and their theoretical analysis for solving complex computational problems.

Fachkompetenzen: Students gain comprehensive knowledge of algorithm design principles and analysis techniques. They learn to develop theoretical solutions for computational problems and prove their correctness using mathematical methods.

Überfachliche Kompetenzen: Students acquire the ability to align theoretical and practical aspects of algorithms. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Advanced algorithmic methods and complexity analysis, including asymptotic analysis and computational complexity theory;
- graph theory and algorithms, structural decompositions, and planarity theory;
- mathematical optimization techniques, including linear programming theory and modelling; and
- advanced algorithm design paradigms and theoretical frameworks, including geometric algorithms, approximation theory, and randomization.

Erwartete Vorkenntnisse: A good understanding of basic algorithms and data structures and methods to analyze them.

These prerequisites are taught in the following modules:

- *Algorithmen und Datenstrukturen*;
- *Effiziente Algorithmen*; and
- *Discrete Mathematics*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Algorithmics

Algorithms for Data Science

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module introduces fundamental algorithms and their analysis for analyzing large amounts of data.

Fachkompetenzen: After successful completion of the module, students are able to

- design and theoretically analyze algorithms for processing large amounts of data,
- implement and use data science algorithms on real-world datasets,
- select an appropriate algorithm for a given data science problem,
- explain how randomization can be exploited to obtain more efficient algorithms

in order to build and analyze highly efficient data processing pipelines for analyzing large amounts of data.

Überfachliche Kompetenzen: Students acquire the ability to understand methods for algorithmically identifying and exploiting properties of data in various domains. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Graph analysis and graph mining (e.g., densest subgraph, arboricity, triangle counting, sublinear-time estimation of Personalized PageRank);
- similarity and nearest neighbor search (e.g., min-hash, locality sensitive hashing);
- streaming algorithms (e.g., heavy hitters, reservoir sampling);
- clustering (e.g., k-Means++, coresets);
- dimensionality reduction (e.g., Johnson–Lindenstrauss lemma, feature hashing, subspace embeddings);
- matrix factorizations and their applications (e.g., singular value decomposition for community detection in random graphs, non-negative matrix factorization, DeepWalk, node2vec).

Erwartete Vorkenntnisse: Knowledge of theoretical algorithm analysis and basics of data science. A good understanding of probability theory and linear algebra is an advantage.

These prerequisites are taught in the following modules:

- *Algorithmen und Datenstrukturen* and
- *Algorithmics*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Algorithms for Data Science

Algorithms in Graph Theory

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with algorithmic aspects of graph theory from a theoretical perspective.

Fachkompetenzen: Students gain the ability to explain basic and advanced concepts and structures in graph theory. They learn to describe and design algorithms for fundamental graph problems while developing theoretical foundations for their analysis.

Überfachliche Kompetenzen: Students acquire the ability to combine structural properties of graphs with algorithm design and analysis. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Matching algorithms and theory;
- planarity testing, embeddings, and algorithms for planar graphs;

- graph width measures and their relationship;
- algorithms for sparse graph classes.

Erwartete Vorkenntnisse: A solid understanding of algorithms, data structures, and graph theory is required.

These prerequisites are taught in the following modules:

- *Algorithmen und Datenstrukturen*;
- *Effiziente Algorithmen*;
- *Discrete Mathematics*; and
- *Algorithmics* (strongly recommended).

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Algorithms in Graph Theory

Applied Generative AI and LLM-based Systems

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module builds on foundational knowledge of generative AI to explore advanced architectural concepts, emerging research directions, and sophisticated integration patterns, with an emphasis on practical implementation and project-based learning.

Fachkompetenzen: After successful completion of the module, students are able to:

- explain internal dynamics, trade-offs, and efficient adaptation (e.g., quantization-aware training, parameter-efficient fine-tuning) of modern transformer and diffusion models;
- apply embedding-based strategies for enhancing generative AI, including conditioning, context augmentation, and retrieval-based improvements; and
- implement, deploy, evaluate, and monitor advanced generative AI systems beyond chatbots for business integration.

Überfachliche Kompetenzen: After the module, students are able to explain the trade-offs between cloud-based and on-premises generative AI solutions, particularly in terms of performance, compliance, and ethical dimensions. They will also be able to assess and communicate the ethical implications of generative AI systems, including transparency, privacy, and bias. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Modern Generative AI Architectures and Paradigms: Foundations of modern generative AI, such as Transformer and Diffusion Architectures. Advanced paradigms including mixture of expert models and diffusion transformers.
- Generative AI in Practice: Advanced model optimization techniques including distillation and pruning strategies, parameter-efficient fine-tuning techniques. Integration/interface patterns of generative AI like augmentation strategies, function calling and tool use, or agentic behavior. Introduction to transparency, explainability, and corrigibility requirements and responsible development practices including output monitoring, content filtering, and safety measures.
- Implementation of Applied Generative AI: Applications of applied generative AI across a range of domains, such as modular text assessment in latent space using LLMs, embedding strategies for retrieving chunked legal documents, privacy-aware document re-purposing, and others.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Applied Generative AI and LLM-based Systems

Artifact-based Design

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with methods and techniques for physical, ubiquitous and artifact-based systems as well as applications thereof.

Fachkompetenzen: After successful completion of the module, students are able to

- Apply iterative design methodologies, including prototyping, testing, and refinement, to create tangible and digital artifacts.
- explain/describe the concepts of Ubiquitous Computing describing the deployment and integration of computing technology into everyday objects and environments.
- explain/describe and apply the relevant concepts and fundamental technologies (e.g. context-awareness, localization and sensing technologies, machine learning on the edge, microprocessor programming, design of ubicomp applications)
- to explain/describe the specific topics of activity recognition, privacy concerns and preserving approaches, business applications, and related business models.
- apply your theoretical knowledge from the class to develop a specific ubicomp project.
- investigation of working principles of sensors and their application for context detection

Überfachliche Kompetenzen: After successful completion of the module, students are able to

- Collaborate effectively in interdisciplinary teams to address complex design problems.
- Reflect on the societal implications of their design work and make informed, ethical decisions.
- Communicate design concepts and processes clearly and persuasively to diverse audiences.
- Adapt to feedback and iterative processes with resilience and a growth mindset.

Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Introduction to artifact-based computing, the history of computing disappearing into the background and serving the user's needs proactively; critical review the vision of ubiquitous computing in the context of today's developments.
- Notion of context-aware computing, the principle of integrating implicit input from user and environment for providing more meaningful services to the user
- Investigation of working principles of sensors and their application for context detection
- Review of location technologies, discussion of privacy-preserving approaches
- Data models and standards for resource-constrained devices
- Revisiting Design, Prototyping & Evaluation Methods
- Review of IoT business models
- Discussion of privacy-related implications
- State of the art of (physical)
- Fabrication technologies
- Outlook on emerging technologies

Erwartete Vorkenntnisse: Knowledge of "Software Engineering" and "Programming"
These prerequisites are taught in the following modules:

- Software Engineering (*Software Engineering*)
- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Artifact-based Design

Artificial Intelligence for Computer Security

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with how artificial intelligence can be used to tackle security-related problems and examines the security challenges posed by AI models.

Fachkompetenzen:

- Students can design and implement suitable features for security tasks, such as malware detection and vulnerability discovery.
- Students can implement and deploy machine learning models tailored to security tasks, such as malware detection, network traffic analysis, or vulnerability assessment.
- Students can identify and analyze common pitfalls, such as data bias and spurious correlations, when evaluating AI-driven systems for security.
- Students can describe and assess the attack surface of AI models.
- Students can apply explainable AI (XAI) techniques to interpret and evaluate the decision-making processes of AI models used in security contexts.

Überfachliche Kompetenzen: Students can analyze the ethical implications of deploying AI in security contexts, including issues of privacy, fairness, and the risk of dual-use in applications like surveillance or cyber-defense.

Inhalt:

- Features and feature spaces for the security domain
- Representation learning in computer security
- Attack detection using anomaly detection techniques
- Attack detection using classification techniques
- Analysis and clustering of malware
- Evaluating learning-based systems for security
- Attacks against learning-based systems
- Privacy threats through AI systems
- Explaining learning-based security systems

Erwartete Vorkenntnisse: The module assumes a foundational understanding of calculus and matrix operations, along with basic programming proficiency.

These prerequisites can be acquired in the bachelor modules *Analysis*, *Algebra und Diskrete Mathematik*, and *Einführung in die Programmierung*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Machine Learning for Computer Security

Automata and Logic

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module explores the deep connections between automata, logic, and algebra, providing a unified framework for analyzing and reasoning about languages, systems, and their properties.

Fachkompetenzen: The focus is on general principles, emphasizing conceptual connections and providing tools for analyzing languages, systems, and their properties through the lenses of automata, logic, and algebra. Students will learn both theoretical foundations and practical methods that have broad applications across computer science and mathematics.

By the end of this module, students will

- understand how automata, logic, and algebra are interrelated and reinforce one another,
- use logic to specify and reason about formal languages and systems,
- apply algebraic structures to analyze and classify automata and languages,
- explore the extension of automata and logic to infinite behaviors and temporal properties, and
- gain insight into how these theoretical tools can be applied in various fields, including verification, synthesis, and computational mathematics.

Überfachliche Kompetenzen: This module fosters interdisciplinary competencies by combining mathematical rigor, logical reasoning, and algebraic techniques to analyze and model computational systems. Students will learn to bridge theoretical and applied perspectives, enabling them to apply automata, logic, and algebraic methods across diverse domains like verification, synthesis, and formal reasoning. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt: The module covers core topics, including:

- finite automata and their relationship to regular languages;
- logical formalisms, such as propositional and monadic second-order (MSO) logic, and their connection to automata;
- algebraic structures (e.g., monoids, semirings) and their role in automata theory and formal languages;
- extensions of automata for infinite behaviors and their connections to temporal logics; and
- applications of automata, logic, and algebra in verification, synthesis, and formal reasoning.

Erwartete Vorkenntnisse: A background in discrete mathematics and a basic understanding of formal languages is recommended.

These prerequisites are taught in the following modules:

- Algebra und Diskrete Mathematik
- Theoretische Informatik

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Automata and Logic

Automated Deduction

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with algorithmic techniques and fundamental results in first-order theorem proving for logic with equality.

Fachkompetenzen: After successful completion of the module, students are able to address the theoretical and practical aspects for devising algorithmic procedures for automated reasoning in first-order logic with equality.

Überfachliche Kompetenzen: Students acquire the ability to understand, apply and use automated deduction methods, in particular within first-order theorem proving tools.

Inhalt:

- Resolution and superposition calculi for first-order logic with equality;
- unification algorithms;
- redundancy checking methods for efficient reasoning;
- saturation-based proof search ;
- inference processes;
- experiments and projects with theorem provers;
- recent advancements in research in theorem proving.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Automated Deduction

Autonomous Racing Cars

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module introduces the students to the hardware, software and algorithms involved in building and racing an autonomous race car.

Fachkompetenzen: After successful completion of the module, students will have the ability to apply and evaluate various autonomous robotic applications.
Überfachliche Kompetenzen: Students acquire the ability to understand methods for developing autonomous-control software and design them themselves for new application domains. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Introduction: Using ROS, the F1/10 hardware, and the F1/10 Simulator.
- Systems: Automatic emergency braking and notions of safety.
- Sensing: LiDAR and rigid body transformations.
- Sensing and Actuation: Reference tracking, Laplace domain dynamics, PID.
- Actuation: Electronic-speed-control tuning.
- Perception I: Localisation by scan matching.
- Perception II: Mapping the world with SLAM and particle filters.
- Planning I: Pure pursuit, a competitive planning method.
- Planning II: Race lines and navigation maps for better control.
- Advanced topics: Rapidly exploring random trees (RRT).
- Advanced topics: Model-Predictive Control (MPC).
- Computer vision: detection, pose estimation and visual feature extraction.
- Machine Learning: Neural network auto-pilots, can a machine learn to drive?
- Learning: Reinforcement-learning and autonomous-vehicles research prototypes.
- End-of-semester F1/10 race with guest teams from other universities.

Erwartete Vorkenntnisse: Stochastic Foundations of Cyber-Physical Systems is a very useful complement.

These prerequisites are taught in the following modules:

- Stochastic Foundations of Cyber-Physical Systems

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Autonomous Racing Cars

Beyond Exact Algorithms

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module introduces randomized and approximation algorithms as efficient alternatives to exact algorithms, focusing on their design, analysis, and applications.
Fachkompetenzen: After successful completion of the module, students are able to

- explain the principles and design goals of randomized and approximation algorithms,
- understand the basic strategies for designing such algorithms,
- design and analyze randomized algorithms with probabilistic performance guarantees,
- design and analyze approximation algorithms with provable solution quality bounds,
- understand the mathematical foundations and formal analysis techniques for the above,
- model computational problems to apply randomized or approximation techniques, and
- investigate new problems and develop novel randomized or approximate solutions.

Überfachliche Kompetenzen: Students are able to perform

- probabilistic reasoning and structured analysis of algorithmic approaches,
- formal reasoning and basic proofs regarding correctness and efficiency,
- abstract modelling of computational problems,
- critical evaluation of trade-offs between accuracy, running time, and resource consumption, and
- being aware of ethical, fairness, and societal issues related to approximate and randomized algorithms.

Inhalt:

- Fundamentals of randomized algorithms: probability theory, expected values, concentration bounds, and others;
- key techniques in randomization such as Monte Carlo methods, random walks, and Markov chains;
- applications of randomization such as sorting, hashing, Bloom filters, linear programming;
- approximation algorithm design techniques such as greedy algorithms, linear programming relaxation, random rounding;
- analysis of approximation ratios and probabilistic guarantees;
- hardness of approximation;
- case studies and practical applications in various domains.

Erwartete Vorkenntnisse: Knowledge of algorithms and data structures, discrete mathematics, basic probability theory, and complexity theory

These prerequisites are taught in the following modules:

- *Algorithmen und Datenstrukturen*;
- *Statistik und Wahrscheinlichkeitstheorie*;
- *Effiziente Algorithmen*; and
- *Algorithmics*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Beyond Exact Algorithms

Business Intelligence

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:
Fachkompetenzen: After successful completion of this module, students will be able to:

- apply analytic methods to extract business insights from vast amounts of data,
- systematically tackle business problems and questions with data,
- compare and contrast the benefits and limitations of various data wareousing architectures,
- design, model, and query a data warehouse,
- define and apply solid data analytics processes to answer analytical questions,
- identify concrete business goals and data mining goals,
- identify the sources of bias in data and apply bias mitigation actions to mitigate discrimination and increase fairness,
- perform solid analyses using both supervised as well as unsupervised machine learning techniques including the necessary preprocessing steps.

Überfachliche Kompetenzen: After successfully completing the module, students will be able to understand the challenges emerging in data-driven decision processes and critically reflect on results obtained and interpret them, specifically considering bias and its impact.

Inhalt:

- Data warehouse reference architecture
- Large-scale data management
- Logical data modeling (star, snowflake)
- Multidimensional data modeling (OLAP)
- Relational modeling
- Querying multidimensional data
- Data analytics process models, specifically CRISP-DM
- Regulatory requirements such as the EU AI Act
- Sources of bias, metrics for measuring bias and bias mitigation
- Preprocessing design decisions and their impact
- Business Intelligence applications

Erwartete Vorkenntnisse: Students are expected to have knowledge in relational database systems.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Business Intelligence

Complexity Theory

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: Complexity theory provides tools to characterize the inherent complexity of problems and, for instance, to formally show that, for a given problem, no significantly more efficient algorithm can be expected.

Fachkompetenzen: After successful completion of the module, students are able to

- explain fundamental complexity classes and their intuition and
- carry out complexity analyses of problems (in particular in the polynomial hierarchy).

Überfachliche Kompetenzen: Students acquire the ability to analyse problems from different domains and to identify the main source(s) of the complexity of a given problem. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt: The module covers basic notions of complexity theory, deterministic und nondeterministic complextiy classes, in particular: the classes L, NL, P, NP, the polynomal hierarchy, PSPACE, and EXPTIME. We also look inside the class P to study parallelizable problems. Students get practice with these complexity classes by analysing the complexity of problems from various applications on various levels of complexity.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Complexity Theory

Computer-Aided Verification

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:

Fachkompetenzen: After successful completion of the module, students are able to

- formally describe systems and specify their correctness using transition relations, automata, and temporal logic,
- understand state-of-the-art model checking algorithms for verification of systems, and
- use contemporary model checking tools and implement a proof-of-concept verification tool.

Überfachliche Kompetenzen: After successful completion of the module, students are able to

- analyze employed techniques and methods,
- select relevant techniques and methods for a given problem, and
- critically assess relevant solutions and formalisms.

Inhalt:

- Modeling of hardware and software;
- specification using temporal logic;
- assertions, and automata;
- explicit-state model checking;
- symbolic model checking with BDDs;
- bounded model checking with SAT;
- abstraction-based algorithms such as interpolation and IC3.

Erwartete Vorkenntnisse: Basic knowledge of theoretical computer science (propositional logic, finite automata); Programming skills.

These prerequisites are taught in modules

- *Theoretische Informatik*
- *Einführung in die Programmierung*

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Computer-Aided Verification

Computer Science Education: Advances in Research and Practice

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with the advancements of computer science education including didactical concepts, learning theories and practice considerations.

Fachkompetenzen: Didactical concepts in Computer Science Education (e.g., inquiry-based/discovery learning, constructionist learning, CS unplugged approaches), designing and evaluating educational interventions and learning activities, research and design methods in Computer Science Education (e.g., design-based research, classroom observations, assessment development and validation), explaining and making computer science concepts understandable to pupils, promoting computational thinking and AI literacy.

Überfachliche Kompetenzen: Exploring a research topic in Computer Science Education by developing new or improve existing educational interventions, including scientific evaluation. Also, leading educational workshops for pupils within the context of TU Wien Informatics eduLAB, interacting with pupils, workshop planning and management, observation and reflection of teaching and learning, team work/collaboration skills, communication/presentation skills. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender and diversity in the context of the Computer Science didactics and education research, particularly focused on extracurricular offers and educational interventions on the topic of Computer Science.

Inhalt: Didactic research and practice skills for educational interventions in the context of TU Wien Informatics eduLAB. Students develop their own research project involving workshop tasks and evaluate them practically with pupils in a workshop-setting. Topics and the used methodological approach to design the learning interventions can be freely chosen by the students, such as algorithms, artificial intelligence, IT security, or logic.

The didactic foundations of the research may include:

- Inquiry-based/Discovery learning
- Constructionist learning
- Scaffolded learning
- Collaborative learning and peer teaching

This module is specifically aimed at students with a deeper interested in making computer science concepts understandable to non-experts, such as pupils. Students can dive into computer science education research and practice, develop and conduct their own didactical research project.

Erwartete Vorkenntnisse: Fundamental Informatics didactics competencies and skills in conducting educational interventions in the context of TU Wien Informatics eduLAB. Knowledge, skills, and competencies from the foundational studies in Informatics, Business Informatics or Computer Engineering, mostly covered by courses in a Bachelor's program.

Verpflichtende Voraussetzungen:

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:

6,0/4,0 PR Computer Science Education: Advances in Research and Practice

Critical Algorithm Studies

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module explores the ethical, social, and political dynamics of algorithmic systems, critically examining their impact on society and the associated challenges.

Fachkompetenzen: After successful completion of the module, students are able to understand fundamental concepts of algorithmic systems, including bias, fairness, transparency, and accountability, as well as strategies to mitigate harms such as misinformation, extremist content, and online harassment.

Überfachliche Kompetenzen: Students gain the ability to critically analyze the societal impact of algorithms, evaluate ethical dilemmas in algorithmic systems, and propose technical or policy-oriented solutions to address trust and safety challenges. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Introduction to critical algorithm studies and trustand safety in algorithmic systems;
- algorithmic imaginaries and future scenarios;
- bias, fairness, and accountability in algorithmic decisions and design;
- technical solutions to harmful online behavior, such as detection and intervention methods;
- ethical and legal frameworks guiding algorithmic systems; and
- applications of trust and safety practices to real-world challenges, including combating misinformation and extremist/terrorist content.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Critical Algorithm Studies

Critical Theory of Media and Informatics

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:

Fachkompetenzen: After successful completion of the module students obtained the following skills:

- Technical and methodological skills:

  - name and explain historical and current day approaches to critical theories as presented in class;
  - assess technologies using critical lenses regarding different axes of power such as gender, class, race, disability, and colonialism.
- Cognitive and practical skills:
  - analyse and discuss the potential societal effects and impacts of technologies;
  - identify and argue for design changes to technological development in front of a technical audience;
  - assess and position academic and general media publications critiquing technologies.
- Social skills and self-competencies:
  - present their assessment in a written manner that is grounded in the literature presented in class;
  - articulate valid critical questions concerning the future of technology in society.

Überfachliche Kompetenzen: Students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Introduction to epistemologies in computer science;
- discussion of different critical theories along different schools of thoughts;
- focussed discussion and analysis of changing topics along current topics relevant to computer science (e.g., emerging technologies and their assessment);
- practical exercises to develop a critical reflexive practice.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Critical Theory of Media and Informatics

Cryptocurrencies

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:

Fachkompetenzen: After successful completion of the module, students are able to describe and analyze the foundations of blockchain technologies, including the basic concepts (mining, consensus protocols, etc.) as well as with the techninques underlying modern cryptocurrencies (layer-2 technologies).

Überfachliche Kompetenzen: Students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Introduction to blockchains and cryptocurrencies;
- consensus in synchrony and partial synchrony;
- longest chain consensus and PoW
- Bitcoin specifics;
- economics of blockchains;
- proof of stake;
- payment channels: the Bitcoin Lightning Network and the state-of-the-art;
- other scaling techniques (e.g., sharding);
- privacy.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Cryptocurrencies

Data Stewardship

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:
Fachkompetenzen: After successful completion of this module, students will be able to:

- explain research data lifecycle
- create a Data Management Plan
- apply FAIR principles in practice
- differentiate between difference license types
- explain what persistent identifiers are
- review existing data management practices
- represent information using machine-actionable DMP
- discuss differences between repository systems
- explain how findability and interoperability of data can be achieved
- describe components of an Research Data Management infrastructure
- describe the scope of RDM policies and how they drive DM activities and obligations
- describe ways to certify a repository and to evaluate certification criteria
- implement reproducible experiments
- explain challenges in digital preservation
- identify significant properties of digital objects and create preservation plans
- explain the role of the OAIS and apply it correctly when developing digital archives

- explain the concepts and processes in trusted research environments for interacting with sensitive or confidential data.

Überfachliche Kompetenzen: After successfully completing the module, students will be able to understand the importance of the role of data, specifically sensitive data (gender, ethnicity, medical data, personal attitudes, etc.) and be aware of processes for determining the ethically correct handling of data in collection and analysis.

Inhalt:

- Reproducible Research
- Machine-actionable Data Management Plans
- Research Data Managenent Services,
- Data Identification and Citation
- Digital Preservation Challenges, Preservation Actions
- ISO OAIS Standard
- Preservation Planning
- Repository Systems
- Repository Certifciation
- Trusted research environments
- FAIR principles, FAIR metrics

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
3,0/2,0 VO Data Stewardship
3,0/2,0 UE Data Stewardship

Database Theory

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:
Fachkompetenzen: After successful completion of the course, students are able to:

- explain the formal concepts of data models and their associated query languages;
- analyze and compare query languages in terms of expressive power and computational complexity;
- apply theoretical results to optimize and evaluate queries systematically; and
- formulate open research questions and critically evaluate new approaches in database theory.

Überfachliche Kompetenzen: Students can formulate open research questions and critically evaluate new approaches in database theory. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt: The module covers key concepts related to the foundations of data management. In particular, the following topics are covered:

- fundamental aspects of database query languages;
- relational query languages (introduction);
- Datalog;
- Codd's Theorem: relational calculus, relational algebra, and Datalog;
- Trakhtenbrot's Theorem;
- complexity of query evaluation;
- conjunctive queries;
- worst-case optimal joins;
- expressive power and Ehrenfeucht-Fraïssé games.

Erwartete Vorkenntnisse: Basic knowledge of database systems, algorithms and data structures, and programming.

These prerequisites are taught in the following module:

- *Formal Methods in Systems Engineering*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Database Theory

Deep Learning for Natural Language Processing

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module introduces students to deep learning approaches for NLP, including hands-on experience with PyTorch.

Fachkompetenzen: After successful completion of the module, students are able to design, implement, and explain neural network models for natural language processing via deep learning, using the PyTorch framework.

Überfachliche Kompetenzen: Students develop the ability to critically assess deep learning applications in NLP, considering ethical challenges and implications. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Word vectors, word window classification, language models;
- backpropagation and neural networks;
- PyTorch framework;
- recurrent neural networks and language models;
- sequence-to-sequence models, machine translation, subword models;
- self-attention and transformers;
- pretraining, natural language generation;
- hugging face transformers;
- prompting, reinforcement learning from human feedback;
- question answering;
- convolutional neural networks, tree recursive neural networks, constituency parsing;
- intersections between NLP and linguistics;
- code generation;
- training large language models;
- multimodal deep learning;
- co-reference resolution;
- interpretability and explainability.

Erwartete Vorkenntnisse: Basic knowledge of programming and machine learning.

These prerequisites are covered in the module *Machine Learning*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Deep Learning for Natural Language Processing

Distributed Systems Technologies

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: The module consists of a lecture part, providing the required theoretical information, and a lab with several programming assignments for an in-depth understanding of the module topics. Furthermore, lab practice lessons will be held for presentation and discussion of the assignments. Moreover, these practice lessons serve to align implementation and the corresponding theory (which is mainly the content of the lecture).

Fachkompetenzen: After successful completion of the module, students are able to implement distributed enterprise applications using appropriate modern distributed systems technologies. They will be able to explain/describe the theory and concepts underlying these technologies, and are therefore able to relate other or new technologies to corresponding problems in distributed systems (e.g., remoting, distributed transactions, caching, API descriptions, messaging, monitoring, or auto scaling). Students will be able to make

informed decisions about which technolgies to use during both design and development phases of distributed enterprise applications.

Überfachliche Kompetenzen: Students acquire the ability to explain/describe methods and technologies for designing and developing enterprise distributed systems applications. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt: Specific concepts explained during the lectures and learned throguth the lab are:

- client-server,
- n-tier systems,
- SQL database abstraction,
- object-related mappings,
- non-relational data models (NoSQL),
- presentation-layer technologies (web framework),
- integration technologies (EAI, web services),
- container technologies (e.g., Docker),
- aspect-oriented middleware,
- message-oriented middleware,
- metaprogramming,
- microservices,
- monitoring.

Erwartete Vorkenntnisse: Knowledge of distributed systems, software engineering, databases, SQL, and object-oriented programming. Good programming skills in Java.

Preceding modules:

- Distributed Systems (*Verteilte Systeme*)
- Database Systems (*Datenbanksysteme*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Distributed Systems Technologies

Efficient Programs

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with the resource consumption of software, whether that is a problem, and if so, how to reduce it.

Fachkompetenzen: Kinds of efficiency (run-time, memory, power). The effect of specification on efficiency. Design for efficiency. Code transformations for increasing efficiency. Usage of tools for understanding resource consumption.

Überfachliche Kompetenzen: Students acquire the ability to determine whether a program is sufficiently efficient (in various respects), to find inefficient parts, and to make them more efficient. They also learn the role of efficient algorithms (constant factors, logarithmic factors), parallel processing, and hardware characteristics. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Which software needs more efficiency? What kind of efficiency?
- Balancing efficiency with other goals in software development
- Methodology for achieving efficiency
- The role of the compiler and programming languages
- The cost of operations: hardware properties and software properties; latency vs. throughput
- The role of algorithms, data structure and parallel processing (mainly discussed in other courses)
- The role of specification and design for efficiency
- Various transformations for improving efficiency
- Energy efficiency
- Tools for understanding where time is spent
- Examples of making software more efficient

Erwartete Vorkenntnisse: Knowledge of "Programming"
These prerequisites are taught in the following modules:

- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
3,0/2,0 VU Efficient Programs
3,0/2,0 PR Efficient Programs

Extension

Regelarbeitsaufwand: up to 12,0 ECTS

Lernergebnisse: This module allows students to extend their profile by choosing courses from other Master curricula that fit the qualification profile of *Software Engineering* .

Fachkompetenzen: Depends on the chosen courses.

Überfachliche Kompetenzen: Depends on the chosen courses.

Inhalt: Depends on the chosen courses.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: The courses in this module can be chosen freely from other Master curricula, provided they fit the qualification profile. Courses from other Informatics Master Curricula at TU Wien can always be chosen.

Fixed-Parameter Algorithms and Complexity

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:

Fachkompetenzen: After successful completion of the module, students are able to understand the theory of parameterized complexity and fixed-parameter tractability in sufficient depth to read and follow latest developments in the area and, crucially, to analyze problems they encounter from the parameterized viewpoint. First and foremost, this includes the ability to obtain asymptotically efficient algorithms and strong lower bounds for problems of interest.

Überfachliche Kompetenzen: Students will learn about the methodology in today's state-of-the-art research on algorithms and complexity, and will be able to better comprehend and produce formal proofs. Students will also be confronted with tasks such as literature research and comparison of known results.

Inhalt: Fixed-parameter algorithms provide a powerful approach for efficiently solving many NP-hard problems by exploiting structural aspects of problem instances in terms of a problem parameter. This module provides an overview of the main techniques for developing fixed-parameter algorithms (including bounded search trees, kernelization, color coding, modulators) as well as the fundamentals of parameterized complexity theory (such as the Weft-hierarchy, XP and para-NP-hardness, kernelization lower bounds) which allows to provide strong evidence that certain problems cannot be solved by a fixed-parameter algorithm.

Erwartete Vorkenntnisse: Students should be aware of the foundations of algorithms and complexity.

These prerequisites are taught in the following modules:

- *Algorithmen und Datenstrukturen*
- *Effiziente Algorithmen*
- *Algorithmics*

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Fixed-Parameter Algorithms and Complexity

Formal Methods for Security and Privacy

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:
Fachkompetenzen: After successful completion of the module, students are able to develop a static analysis technique to enforce security and privacy properties in a variety of domains, such as cryptographic protocols, programming languages, bytecode, and deep neural networks. In particular, this module explains the foundations of the static analysis of security and privacy properties, with a particular focus on SMT solving and type systems. Students will learn to formalize a static analysis, prove its soundness, and implement it in an efficient way using state-of-the-art verification tools.
Überfachliche Kompetenzen: Students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Foundations of theorem proving;
- type theory and static analysis for security and privacy;
- formal modelling and verification of security and privacy in software;
- cryptographic protocols, Blockchains, and machine learning.

Erwartete Vorkenntnisse: Foundations of security and logic taught in typical bachelor's programs in computer science.

These prerequisites are taught in the following bachelor modules:

- *Einführung in Security*;
- *Logic and Reasoning in Computer Science*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Formal Methods for Security and Privacy

Formal Methods in Systems Engineering

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with foundations and applications of automated reasoning and verification techniques for computer systems.

Fachkompetenzen: After successful completion of the module, students are able to understand the theory and practice of automated reasoning and verification techniques for computer systems.

Überfachliche Kompetenzen: Students acquire the ability to understand, use, and apply formal methods for ensuring system correctness.

Inhalt:

- Methods for checking correctness of system requirements, in particular using SAT and satisfiability modulo theory (SMT) solving;
- model checking approaches, in particular using temporal logics and bounded model checking;
- deductive verification techniques, such as Hoare logic and weakest precondition reasoning;
- program-analysis methods, in particular interval analysis and pointer semantics.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Formal Methods in Systems Engineering

Freie Wahlfächer und Transferable Skills

Regelarbeitsaufwand: 9,0 ECTS

Lernergebnisse: Die Lehrveranstaltungen dieses Moduls dienen der Vertiefung des Faches sowie der Aneignung außerfachlicher Kenntnisse, Fähigkeiten und Kompetenzen.

Inhalt: Abhängig von den gewählten Lehrveranstaltungen.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: Die Lehrveranstaltungen dieses Moduls können frei aus dem Angebot an wissenschaftlichen und künstlerischen Lehrveranstaltungen, die der Vertiefung des Faches oder der Aneignung außerfachlicher Kenntnisse, Fähigkeiten und

Kompetenzen dienen, aller anerkannten in- und ausländischen postsekundären Bildungseinrichtungen ausgewählt werden, mit der Einschränkung, dass zumindest 4,5 ECTS aus den Themenbereichen der Transferable Skills zu wählen sind. Für die Themenbereiche der Transferable Skills werden insbesondere Lehrveranstaltungen aus dem Wahlfachkatalog „Transferable Skills“ der Fakultät für Informatik (Anhang E) und aus dem zentralen Wahlfachkatalog der TU Wien für „Transferable Skills“ empfohlen.

Generative AI

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module provides a comprehensive introduction to modern generative AI fundamentals and applications, from large language models (LLMs) to multimodal systems, including practical implementation aspects, intersections with related research areas, and ethical considerations.

Fachkompetenzen: After successful completion of the module, students are able to:

- explain the fundamentals of generative AI, including language models, multimodal approaches (vision transformers, diffusion models), and the transformer architecture;
- apply training techniques such as pre-training, fine-tuning, and reinforcement learning;
- integrate LLMs using advanced prompting, retrieval-augmented generation, and tools like LangChain;
- assess the role, potential, and limitations of generative AI in applications like recommender systems;
- use knowledge graphs to address generative AI challenges such as hallucinations and lack of domain knowledge; and
- explain the intersection between symbolic AI and subsymbolic architectures in generative models.

Überfachliche Kompetenzen: Students learn to critically evaluate the ethical implications of generative AI systems, their potential biases and societal impact. Moreover, they learn the intersections between generative AI and other research areas like recommender systems or symbolic AI systems. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Theory of LLMs and generative AI:
  - AI paradigms and generative AI within the broader taxonomy, covering autoregressive language modeling, probabilistic approaches, tokenization, and embedding spaces;
  - Transformers, encoder-decoder structures, self-attention mechanisms, and LLM training, including reinforcement learning and proximal policy optimization.

- Practice of LLMs and generative AI:
  - advanced prompt engineering to address LLM limitations;
  - retrieval-augmented generation (RAG), including architecture, integration, and enhancement techniques;
  - tooling for LLM deployment, interaction, and integration.
- Generative AI beyond language modelling:
  - multimodal AI, including vision transformers, text-to-speech models, and diffusion models;
  - knowledge graphs (KGs) for addressing LLM limitations (hallucinations, domain knowledge, trust) using KG embeddings, graph neural networks, and GraphRAG;
  - symbolic and subsymbolic AI intersections, knowledge-enhanced generation, and LLM-enhanced knowledge representations;
  - generative AI in recommender systems, including architecture, evaluation, and limitations;
  - ethical considerations, covering bias, fairness, privacy, and embedded ethical frameworks.

Erwartete Vorkenntnisse: Basic knowledge of programming and machine learning.

These prerequisites are taught in the module *Machine Learning*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Generative AI

GPU Computing and Architectures

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: After successful completion of the module, students are able to master GPU-based architectures and related technologies and to develop efficient parallel algorithms based on many-cores. Through the final assignment, they will also acquire important team working skills.

Fachkompetenzen: The module covers the fundamentals of GPU architecture and hardware, emphasizing many-core designs and the principles of parallel execution. Students learn the basics of CUDA programming, including threading models, execution configurations, and memory hierarchies, and develop skills in optimizing data transfers and leveraging shared, constant, and global memory spaces. They explore parallel patterns such as map, reduce, and scan, while also examining advanced features like dynamic parallelism and concurrency through CUDA streams. This knowledge is reinforced by studying profiling

techniques, optimization strategies, and the use of the Thrust library for high-level parallel primitives. Throughout the module, students apply their learning by managing projects in a collaborative GitHub environment, enabling them to track progress, coordinate tasks, and document their work effectively.

Überfachliche Kompetenzen: The module fosters interdisciplinary competencies by engaging students in team-based project work that requires collaborative problem-solving, effective communication, and project management skills. Through regular progress reviews and presentations, students learn to plan, delegate tasks, and adapt to changing objectives within a group setting. They also develop the ability to critically assess the feasibility and risks of complex technical solutions, encouraging a blend of engineering precision and strategic thinking. Furthermore, embedded ethics is addressed by guiding students to consider the social and environmental impact of GPU-accelerated computing. They examine resource efficiency, responsible data handling, and potential biases in algorithmic implementations, ensuring that technological innovation aligns with broader ethical standards and sustainability goals.

Inhalt: This is the list of the main topics of the module:

- GPU Architectures
- CUDA Programming (Basics)
- GPU Memories/Access Patterns
- CUDA Parallel patterns
- Thrust Library
- Dynamic Parallelism
- CUDA application profiling
- CUDA Streaming

Erwartete Vorkenntnisse: Students entering this module are expected to possess a solid foundation in programming (preferably with C/C++), a basic understanding of computer architecture, and familiarity with fundamental concepts of parallel computing or multithreading. Prior experience using version control systems like Git, as well as an ability to collaborate within a team setting, will also help students more quickly adapt to the course's practical assignments and project-based learning environment.

These prerequisites are taught in the following modules:

- Digital Design and Computer Architecture (*GPU Computing and Architectures*)
- Operating Systems (*Betriebssysteme*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU GPU Computing And Architectures

Graph Drawing Algorithms

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module introduces efficient algorithms and complexity results for drawing and representing graphs in different visual layout styles.

Fachkompetenzen: Students are able to

- explain fundamental concepts, structures, and problems in graph drawing,
- explain and compare different aesthetic optimization goals in graph drawing,
- design and analyze graph drawing algorithms,
- model graph drawing problems and adapt known algorithms and graph layout styles to related problems,
- implement and evaluate graph drawing algorithms, and
- investigate new graph drawing problems and develop new layout algorithms.

Überfachliche Kompetenzen: Student are able to perform

- formal reasoning, correctness proofs, and proofs of computational complexity,
- structured analysis of computational geometric graph representation problems,
- modeling and solving interdisciplinary problems at the interface of application domains, information visualization, design, and graph algorithms,
- to judge trade-offs between different conflicting optimization goals, and
- being aware of possible ethical, gender, and diversity issues in the context of network visualization

Inhalt: The module content covers fundamental and advanced topics in graph drawing including

- graph layout for restricted graph classes such as trees, planar graphs, directed graphs,
- optimization problems for different quality metrics and drawing styles in graph drawing,
- general purpose algorithms using physical analogies,
- graph drawing frameworks, and
- heuristics/approximation algorithms for NP-hard problems.

Erwartete Vorkenntnisse: Extended knowledge of algorithms and basics of graph theory.

These prerequisites are taught in the following modules:

- *Effiziente Algorithmen*;
- *Discrete Mathematics*; and
- *Algorithmics*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Graph Drawing Algorithms

Green HPC

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: Lectures explore the basic energy impact of HPC, highlighting the immense power demands of modern data centers and their contribution to global energy consumption and CO2 emissions.

Fachkompetenzen: Students will gain an understanding of how HPC infrastructures affect sustainability and why energy efficiency is now a critical focus for the field. Next, the lecture explores sustainability measures and key performance indicators (KPIs) used in state-of-the-art HPC data centers. These include metrics for energy efficiency (e.g., Power Usage Effectiveness, or PUE), renewable energy integration, and waste heat recovery systems. Students will learn how these measures are implemented in cutting-edge facilities to minimize environmental impact while maintaining high computational performance. The lecture then delves into methods and tools for assessing energy consumption and CO2 footprints of diverse HPC applications, such as large language models (LLMs) and federated learning. It examines tools like energy monitors and simulation frameworks, showcasing how to profile and optimize applications to improve their energy efficiency. Real-world examples will be discussed, providing insights into the environmental costs of AI-driven workloads. Finally, the lecture covers green workload scheduling algorithms, which aim to optimize resource utilization while minimizing energy consumption. Students will explore techniques like workload shifting in space (e.g., distributing tasks across energy-efficient nodes) and time (e.g., scheduling tasks during periods of low carbon intensity in the grid). These strategies are essential for reducing operational costs and environmental impact while maintaining system performance.

Überfachliche Kompetenzen: Students will develop a range of interdisciplinary skills from the Green HPC lecture, integrating expertise across computer science, engineering, environmental science, and policy. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Anatomy of HPC infrastructure: nodes, storage, networks, and cooling
- Sustainability Measures and KPIs in Data Centers e.g. Power Usage Effectiveness (PUE), Carbon Usage Effectiveness (CUE), and Energy Reuse Factor (ERF).
- Frameworks for measuring the carbon footprint of HPC workloads

- Profiling energy demands of applications like LLMs and Federated Learning
- Workload-shifting techniques in space (efficient node usage) and time (carbon-aware scheduling).

Erwartete Vorkenntnisse: Proficiency in a programming language commonly used in HPC, such as Python, basic understanding of HPC architectures, including concepts like distributed computing, parallel processing, and hardware components (e.g., CPUs, GPUs, TPUs).

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Green HPC

Heuristic Optimization Techniques

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:

Fachkompetenzen: After successful completion of the module, students are able to, students understand concepts and techniques from heuristic optimization and problem solving, and are able to apply them in efficient implementations as well as to systematically evaluate and compare approaches experimentally.

Überfachliche Kompetenzen: Students acquire the ability to understand, develop, apply, and experimentally evaluate algorithms for heuristically solving challenging computational optimization problems in practice. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Construction heuristics;
- local search techniques;
- metaheuristics, including simulated annealing, GRASP, tabu search, variable neighborhood search, evolutionary algorithms, and ant colony optimization;
- hybrid optimization techniques, including large neighborhood search techniques;
- Mmchine learning based approaches;
- parallelization;
- analysis and tuning.

Erwartete Vorkenntnisse: Solid programming skills and knowledge in algorithms and data structures

These prerequisites are taught in the following modules:

- *Algorithmen und Datenstrukturen*;
- *Einführung in die Programmierung.*

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Heuristic Optimization Techniques

High Performance Computing

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: Knowledge of current state of affairs in High Performance Computing systems and architectures, skills in advanced usage of programming frameworks and tools for HPC, algorithmic paradigms for basic communication problems.

Fachkompetenzen: Architectures and systems for High Performance Computing, communication problems, algorithms under different network and system assumptions, problems in benchmarking and substantiating performance claims. Advanced aspects of programming frameworks like MPI.

Überfachliche Kompetenzen: Students will gain a deeper understanding of performance of, typically, large, complex, highly parallel computing systems, and be able to better appreciate challenges in efficiently exploiting such systems.

Inhalt: After successful completion of the module, students are able to

- Assess characteristics of High-Performance Computers
- Assess expected performance of parallel programs
- Use advanced features of the MPI standard
- Use advanced features of the OpenMP standard
- Understand, analyze, and design algorithms for communication operations

Erwartete Vorkenntnisse: Basic knowledge in parallel computing, computer architecture, software engineering, programming (in C or related languages) and algorithms.

These prerequisites are taught in the following modules:

- Parallel Computing (*Parallel Computing*)
- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU High Performance Computing

HPC for AI

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: Students will gain an understanding of the challenges and methods for scaling AI in HPC with highlights on practical use cases that demonstrate the integration of AI with HPC infrastructure.

Fachkompetenzen: Lectures discuss key obstacles, including the rapid advancements in hardware (e.g., GPUs, TPUs, quantum computing) and its implications for AI workloads. Further, we will discuss the exponential growth of data, which demands innovative solutions for efficient AI integration in HPC environments. A significant focus is placed on methods for modeling, implementing, and verifying tradeoffs between accuracy and performance, emphasizing the need to optimize computational efficiency while maintaining precision. Techniques for dynamic workload optimization in HPC settings (e.g., tuning hyperparameters, quantization, and pruning in large-scale AI models). The lecture also delves into techniques for scaling large-scale AI systems, particularly in model building and inference, while addressing challenges such as hardware heterogeneity, distributed training, and workload optimization. Regarding hardware heterogeneity, we will discuss integrating FPGAs and other accelerators. Practical application will be discussed, including climate modeling and large language models (e.g., from training massive datasets to optimizing inference latency). We will highlight the real-world impact of AI in HPC, demonstrating approaches to runtime optimization and resource management for complex systems.

Überfachliche Kompetenzen: Students taking this module will gain a range of interdisciplinary skills that blend knowledge from computer science, engineering, and data science, preparing them to tackle complex challenges. Students will learn to analyze and design solutions at the intersection of hardware, software, and data systems, enabling them to understand how AI and HPC interact holistically in large-scale environments. They will acquire the ability to evaluate tradeoffs between accuracy, performance, and resource efficiency, a critical skill in balancing computational demands with practical constraints like energy consumption. Weiters können die Studierenden ethische Fragestellungen im Kontext der Inhalte des Moduls identifizieren, formulieren und diskutieren.

Inhalt:

- Challenges in scaling AI for HPC environments
- Methods for balancing accuracy, performance, and resource efficiency
- Rapid evolution of hardware (e.g., GPUs, TPUs, quantum computing) and its implications for AI workloads.
- Techniques for dynamic workload optimization in HPC settings.
- Dealing with hardware heterogeneity (e.g., integration of FPGAs, TPUs, and other accelerators).
- Algorithmic improvements for distributed model training and inference.

Erwartete Vorkenntnisse: Basics of machine learning, proficiency in a programming language commonly used in AI and HPC, such as Python, basic understanding of HPC

architectures, including concepts like distributed computing, parallel processing, and hardware components (e.g., CPUs, GPUs, TPUs).

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU HPC for AI

Human-agent Interaction

Regelarbeitsaufwand: 12,0 ECTS

Lernergebnisse:
Fachkompetenzen: After successful completion of the course, students are able to:

- Describe foundational concepts in Human-Agent Interaction, including embodiment, anthropomorphism, socio-morphism, trust, and authenticity.
- Critically evaluate the effectiveness of agent interactions through user research methods.
- Synthesize findings on human perception of agency to create conceptual prototypes and define detailed system requirements.
- Analyze human perceptions of and responses to agents across different contexts.
- Plan, execute, and interpret user studies of multimodal Human-Agent Interaction.
- Communicate design processes and evaluation findings to diverse audiences.

Überfachliche Kompetenzen:

- Recognize and explain the impact of design cues that foster apparent agency.
- Evaluate and reflect on the societal and ethical implications of integrating technology with apparent agency into everyday life.

Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Introduction to HAI: Overview of intelligent systems with agency and their integration into daily life.
- Key Concepts: Embodiment, anthropomorphism, socio-morphism, agent design, multimodal communication, trust, and authenticity.
- Human Perception and Emergent Interaction: Understanding how humans perceive and interact with agents in various roles (e.g., assistants, teammates, companions).
- Designing for HAI: Principles for creating engaging, effective, and human-centered interactions.

- Evaluation Methods: Utilize and adapt diverse user research methods to evaluate Human-Agent Interaction in terms of usability, acceptance, engagement, and likability.
- Application Contexts: Real-world uses of HAI in fields such as healthcare, education, customer service, and entertainment.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6/4,0 VU Human-agent Interaction
6,0/4,0 PR Human-agent Interaction

Hybrid Quantum - Classical Systems

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: After successful completion of the module, students are able to understand quantum computing methods and apply them for the development of hybrid applications, i.e., applications that consist of classic and quantum tasks, and apply them to typical scientific applications.

Fachkompetenzen: Fundamental concepts and techniques on hybrid classic quantum systems

Überfachliche Kompetenzen: Students acquire the ability to understand and use methods to assess and utilized hybrid classic quantum systems and implication of such systems on a broader scale Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Concepts of Hybrid Systems
  - Introduction to HPC
  - Connection to Quantum
  - Hybrid System
  - First Assignment
- Basics of Quantum Computing
  - Vectors, Matrices, dirac notation, tensor product(inner product),
  - Qubit, measuring qubit, visualizing qubit, single qubit & multiple qubit states, Qiskit
  - Pure states, mixed states. entaglements
  - Second Assignment

- Hybrid Classic/Quantum Programming
  - Quantum circuit, operators, gates, simulating on local machine or cloud
  - Time evolution of quantum system
  - Variational Quantum Algorithms
  - Third Assignment

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Hybrid Quantum - Classical Systems

Information Technology in Automation

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: One of the biggest challenges for connecting industrial automation systems with IT enviroments via Internet of Things technologies comes from the inherent technical complexity of the underlying systems. The content of the module is the transfer of knowledge for linking and merging different technology areas (including embedded systems and cloud computing) via standardized (Internet-based) interfaces. As part of the module, different approaches to information and application modeling in the industrial environment are discussed and challenges for the implementation of the Industrial Internet of Things (IIoT) are identified. In the accompanying exercise part, proof-of-concepts are created, documented in laboratory protocols and explained in submission discussions.

Fachkompetenzen: By the end of this module, the student will be able to evaluate Information models and communication protocols used in Industrial Internet of Things applications.

Überfachliche Kompetenzen: Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Structure and organisation of distributed automation systems
- In-depth study of communication protocols and middlewares used in automation systems
- Current developments in the Industrial Internet of Things
- Information models as the basis for integrating heterogeneous systems
- Machine-to-machine communication and device management

Erwartete Vorkenntnisse: Object-oriented programming skills and basic structure of computer systems.

These prerequisites are taught in the following modules:

- Programming (*Einführung in die Programmierung*)
- Computersysteme

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Information Technology in Automation

Internet of Things

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: After successful completion of the module, students are able to design and develop an IoT.

Fachkompetenzen: Students should be able to solve, within a team, the common related problems. They should master the knowledge of IoT protocols and they should be able to assess both ethical and security issues arising in the context of IoTs.

Überfachliche Kompetenzen:
Interdisciplinary competencies:

- Systems Thinking
- Data Analytics
- Project Management
- User-Centered Design

Embedded Ethics:

- Privacy: implications of data privacy
- Security: study the potential attacks of a hacker trying to get access to IoTs
- Sustainability: environmental impact of producing and disposing of IoT devices

Weiters können die Studierenden ethische Fragestellungen im Kontext der Inhalte des Moduls identifizieren, formulieren und diskutieren.

Inhalt:
The main topics are:

- Smart Things as Cyber-Physical Systems
- Radio Frequency Identification
- Wireless Sensor Networks (ZigBee)
- Low energy Bluetooth communication
- IoT protocols (MQTT, REST, CoAP, etc)
- Fog and Edge Computing

- Security and Privacy of Internet of Things

Erwartete Vorkenntnisse:

- Programming Knowledge: Proficiency in C, C++ and Python programming languages.
- Basic Electronics: Understanding of basic electronic concepts, including reading schematics, using breadboards, and basic components like resistors, capacitors, and microcontrollers.
- Knowledge of basic networking concepts
- Understanding of Operating Systems
- Problem-Solving Skills
- Critical Thinking

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: We will provide some fundamental practical experience, giving real examples and demonstrations. The lectures will be held all in presence. The examination consists of a project in a group of a maximum of three people and a final presentation of the results.

Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Internet of Things

Introduction to Computational Sustainability

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: Modern ICT systems, such as data centers, AI models, and high-performance computing (HPC), consume significant amounts of energy and resources, contributing to growing environmental concerns. As global sustainability goals, including the UN's Sustainable Development Goals (SDGs), become increasingly urgent, computing emerges as a powerful tool to address challenges in areas such as climate modeling, renewable energy optimization, and resource management. Computational sustainability extends beyond traditional computing by equipping students with the knowledge and skills to design algorithms that balance efficiency, cost, and environmental impact, enabling them to develop innovative, sustainable solutions for complex problems.

Fachkompetenzen: Fundamental concepts and techniques on computational sustainability.

Überfachliche Kompetenzen: Students acquire the ability to describe and explain methods for computational sustainability and assess and evaluate existing systems regarding their sustainability. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Sustainable AI: Impact on sustainability by AI models
- Hardware advancements, data explosion, and its energy impacts
- Energy Challenge of AI models: Cost of Training and Inference
- Large Language Models and their energy consumption
- Methods to address energy consumption of AI models
- AI for sustainability: Using AI to combat the climate change issues
- Specific Use cases on AI for sustainability

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Introduction to Computational Sustainability

Knowledge Graphs

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:

Fachkompetenzen: Learning outcomes are divided into three main blocks: (i) *representations of knowledge graphs* (logic- and ML-based), (ii) *systems for knowledge graphs* (scalablility and reasoning), and (iii) *applications of knowledge graphs* (real-world enterprise AI). An overarching aim of the module is to understand the connections between knowledge graphs (KGs), artificial intelligence (AI), machine learning (ML), deep learning, and data science.

Überfachliche Kompetenzen: Students are able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content. This relates particularly to the role of knowledge graphs as a way to validate information and knowledge in AI systems as well as the ability of many of the knowledge graph techniques discussed in the module to provide explainable AI solutions – hence a key part of providing ethical and explainable AI.

Inhalt: The module includes the following topics:

- knowledge graph embeddings;
- logical knowledge in knowledge graphs;
- graph neural networks;
- graph transformers;
- knowledge graph architectures;
- scalable reasoning in knowledge graphs;
- the knowledge graph lifecycle (creation, evolution, services);
- real-world applications of knowledge graphs (including financial applications).

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Knowledge Graphs

Learning Technologies and Learning Analytics

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: The module covers foundations, research trends, and applications in the field of learning technologies and learning analytics.

Fachkompetenzen: Concepts of learning technologies and different learning environments; emerging technologies for teaching and learning (AI, XR, serious games, gamification); instructional design theory and frameworks; educational data mining and learning analytics; ethical, legal and social aspects in learning technologies research and practice.

Überfachliche Kompetenzen: Project management; team work/collaboration skills; communication/presentation skills; information visualization; Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender and diversity in the context of the learning technologies and learning analytics, particularly focused on technology design and usage.

Inhalt: The course covers various topics in the research areas of learning technologies and learning analytics. It takes a human-centered perspective and focuses on the educational domain, its user groups and contexts. Topics may include:

- Learning Theories
- Learning Environments
- Learning Technologies, e.g. Web, Mobile, Games, VR/AR
- Learning Analytics Infrastructures
- Learning Analytics Methods
- AI in Education
- E-Assessment
- Ethics, Gender and Diversity in Education

Besides regular lectures introducing the different topics, students will work on a collaborative project designing and implementing a prototype involving learning technologies and learning analytics. Topics for projects are proposed by the instructors, but students are also encouraged to submit their own original ideas to work on.

Erwartete Vorkenntnisse: Foundations of software engineering and data science, Programming skills, interest in learning and teaching as application areas in computer science

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Regular lectures and assignment sessions, project work incl. deliverables, group presentations, and oral examination

Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Learning Technologies and Learning Analytics

Logic and Computability

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: The module is concerned with a broad range of logical formalisms as modelling and specification tools as well as with techniques for automated proof search. The aim is to provide a solid understanding of the role of mathematical logic and of computability theory in computer science.

Fachkompetenzen: After successful completion of the module, students are able to apply logical formalisms in various application scenarios and to employ concepts of formal logic and computability theory as tools for effective, precise, and innovative problem solving in IT.

Überfachliche Kompetenzen: The students gain logical, mathematical, and computational knowledge and abilities. Moreover, they also understand philosophical aspects relating to the incompleteness of proof systems and to the interpretation of logical connectives.

Inhalt: The topics in logic comprise the following:

- advanced aspects of classical first-order logic as a specification tool;
- expressibility (elements of model theory);
- proof systems for classical first-order logic (including soundness and completeness proofs);
- a comparison of different types of inference systems,
- methods for handling identity;
- elements of modal logic (including temporal and epistemic logics);
- elements of intuitionistic logic and constructivism; and
- principles of automated theorem proving.

As for topics about computability, these include:

- different models of computations;
- the Church-Turing thesis;
- decidable and undecidable problems alongside formal tools to distinguish them;
- various concepts of problem reductions;
- the incompleteness of arithmetic and its consequences for the verification of programs, and

- connections between computability and logic.

Erwartete Vorkenntnisse: Students should have a basic knowledge of classical propositional and first-order logic, of different programming paradigms (imperative, functional, logical), of concepts of formal languages (grammars, Chomsky hierarchy), and of complexity theory. Moreover, students should be able to express problems and requirements in a precise mathematical language.

These prerequisites are taught in the following modules

- *Formal Methods in Systems Engineering*;
- *Theoretische Informatik*; and
- *Algebra und Diskrete Mathematik*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Logic and Computability

Logic-based Artificial Intelligence

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: At the end of this module, the students will be familiar with a range of logic-based formalisms for describing problem domains. They will be able to select an adequate formalism for a problem of interest, and correctly model the problem in the selected formalisms. The student should also be able to select an adequate reasoning engine for solving the problem at hand, informed by an understanding of the problems it implements.

Fachkompetenzen: The students who complete the module are able to

- read and write correctly domain specifications in classical predicate logic,
- decide whether a given statement is consistent with or entailed by a logic specification,
- describe a given domain using a description logics of adequate expressiveness,
- understand and solve instances of classical DL reasoning services: consistency, instance checking, and concept subsumption,
- understand the principles of nonmonotonic reasoning and recognise different nonmonotonic entailment relations,
- correctly write logic programs witdh default negation and compute their answer set semantics,
- model probabilistic domains and draw inferences from them,
- recognise different types of decision rules, and

- apply algorithms to learn rules from tabular data.

Überfachliche Kompetenzen: Students are able to model problem domains accurately, and to assess and criticize the adequacy of different technqiues for specific situations. The students can identify and verbalise ethical challenges that arise when applying AI techniques in the real world.

The material taught in this module does not involve gender aspects. Teaching resources and methods will be developed with an awareness of diversity and ensuring a welcoming environment for students from underrepresented groups.

Inhalt: The module discusses the following topics:

- propositional and predicate logic as knowledge representation languages;
- description logics and ontological reasoning;
- tractable ontological reasoning;
- foundations of nonmonotonic reasoning;
- answer-set programming and declarative problem solving;
- logics with uncertainty;
- rule learning; and
- applications.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Logic-based Artificial Intelligence

Low-Level Programming

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with low-level programming for parts of software beyond the reach of high-level programming or where high-level programming is too inefficient.

Fachkompetenzen: Memory organization in various kinds of software. Memory safety. Programming in Forth. Programming in Rust. Methods and tools, in particular for debugging.

Überfachliche Kompetenzen: Security problems of low-level programming. Systems programming. Run-time systems for high-level programming languages. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- The role of low-level programming in the software ecosystem

- Limits of high-level programming: operating systems, drivers, run-time systems, efficiency
- Various forms of memory allocation (static, explicit deallocation, automatic deallocation)
- Security and reliability problems of low-level programming
- Programming in Forth (interactive low-level programming)
- Programming in Rust (mostly-safe low-level programming)
- Methods and tools, in particular for debugging

Erwartete Vorkenntnisse: Knowledge of "Programming"

- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Low-Level Programming

Machine Learning

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: Students will learn fundamental machine learning concepts and techniques and apply them in different domains.

Fachkompetenzen: After successful completion of the module, students are able to:

- explain fundamental concepts of machine learning;
- explain the basics of machine learning theory;
- formulate problems as specific machine learning tasks;
- explain and implement the main supervised machine learning algorithms;
- explain and implement basic clustering algorithms;
- explain and implement basic reinforcement learning algorithms;
- apply learning techniques to various datasets and application domains;
- explain and apply automated machine learning methods;
- evaluate and compare machine learning methods; and
- explain and apply data preprocessing techniques.

Überfachliche Kompetenzen: Students acquire the ability to assess and use machine learning in different application domains. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content. This includes the problem of bias that arises in datasets and learning algorithms, the fairness of learning algorithms with respect to gender, race, age, and

other factors, the transparency of machine learning systems, and the societal impact of automated decision-making based on machine learning.

Inhalt: The module includes the following topics:

- fundamental concepts of machine learning;
- basics of machine learning theory;
- fundamental supervised machine learning techniques, including basic techniques and advanced methods such as deep learning;
- basic clustering algorithms;
- basics of reinforcement learning and main tabular methods;
- evaluation and comparison of machine learning techniques;
- data preprocessing techniques and feature selection methods; and
- basic concepts of automated machine learning.

Erwartete Vorkenntnisse: Knowledge of programming, mathematics, and basic algorithms.

These prerequisites are covered in the following modules:

- *Einführung in die Programmierung*;
- *Algebra und Diskrete Mathematik*; and
- *Statistik und Wahrscheinlichkeitstheorie*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Machine Learning

Machine Learning for Optimization

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: Students will learn fundamental concepts of solving techniques based on machine learning and apply them to various combinatorial optimization problems.

Fachkompetenzen: After successful completion of the module, students are able to:

- explain fundamental concepts of combinatorial optimization problems;
- explain concepts of end-to-end learning;
- explain and implement approaches based on deep learning and graph neural networks to solve combinatorial optimization problems;
- explain and implement reinforcement learning-based approaches for combinatorial optimization;
- apply large language models to combinatorial optimization;

- explain how machine learning can be used to improve general-purpose solvers.

Überfachliche Kompetenzen: Students develop the ability to critically assess the benefits of machine learning methods when solving optimization problems. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content. This includes the problem of bias that arises in datasets and learning algorithms, the fairness of learning algorithms with respect to gender, race, age, and other factors, the transparency of machine learning systems, and the societal impact of automated decision-making based on machine learning.

Inhalt: The module includes the following topics:

- combinatorial optimization problems;
- end-to-end learning for combinatorial optimization;
- deep learning and graph neural networks for combinatorial optimization;
- reinforcement learning for combinatorial optimization;
- large language models for combinatorial optimization; and
- machine learning for general-purpose solvers.

Erwartete Vorkenntnisse: Knowledge of programming and machine learning techniques.

These prerequisites are covered in the following modules:

- *Machine Learning*; and
- *Einführung in die Programmierung.*

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Machine Learning for Optimization

Management of Graph Data

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module provides a comprehensive introduction to the principles, technologies, and advanced topics in graph data management, essential for working with knowledge graphs and different graph data models.

Fachkompetenzen: After successful completion of this module, students will be able to:

- Explain key graph data models and query languages,
- Formulate complex queries using graph query languages,
- Explain principles of query optimization,

- Apply schema languages tailored for graph data,
- Ensure data quality through shape constraints,
- Apply and model provenance techniques,
- Manage dynamic updates,
- Use state-of-the-art graph data technologies in popular applications and use cases.

Überfachliche Kompetenzen: Students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt: The module covers essential concepts and techniques for effective graph data management, including:

- Key concepts of graph data models and schema languages
- Query formulation, execution, and optimization
- Ensuring graph data quality through validation and shape constraints
- Methods for tracking provenance and ensuring traceability of graph data
- Strategies for managing dynamic updates and versioning of knowledge graphs
- Recent advances and trends in graph data technologies

Erwartete Vorkenntnisse: Students are expected to have basic knowledge in relational database systems and algorithms and data structures.

These prerequisites are taught in the following modules:

- *Datenbanksysteme*
- *Algorithmen und Datenstrukturen*

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Management of Graph Data

Mathematical Programming and Optimization in Transport Logistics

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with solving optimization problems arising in transport logistics and related areas with exact and heuristic methods. A main focus is on mathematical programming techniques, in particular integer linear programming as well as the theoretical and technical background for doing so successfully.

Fachkompetenzen: After successful completion of the module, students are able to

- understand theoretical foundations of linear and integer linear programming (ILP),
- model optimization problems as ILP formulations,

- compare different ILP formulations for optimization problems theoretically and in practice,
- apply methods and algorithms for solving ILP models in general, and
- use specific ILP formulations and algorithms (exact and heuristic) for solving various optimization problems in transport logistics and related areas.

Überfachliche Kompetenzen: Students learn to

- apply the aforementioned techniques to problem settings from various fields,
- work together in groups on these problems, and
- present their work as a technical report.

Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt: This module covers

- fundamentals of linear and integer linear programming (ILP) and its place within mathematical optimization,
- ILP modelling and solving techniques, both basic (compact formulations) and advanced (branch-and-cut, branch-and-price, Lagrangian relaxation, decomposition methods),
- ILP theory (analysis of different formulations, valid inequalities),
- optimization problems from transport logistics, such as the travelling salesperson problem, the vehicle routing problem and the pickup and delivery problem, and other related areas
- exact and heuristic methods for solving these, and
- real-world optimization problems and their complexities.

Erwartete Vorkenntnisse: Students are expected to be familiar with

- programming,
- linear algebra,
- graph theory,
- fundamentals of (integer) linear programming, and
- fundamentals of heuristic optimization.

These prerequisites are taught in the following modules:

- *Einführung in die Programmierung*
- *Algebra und Diskrete Mathematik*
- *Algorithmics*
- *Heuristic Optimization Techniques*

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Mathematical Programming and Optimization in Transport Logistics

Mobile Robotics

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module covers fundamental concepts and techniques in mobile robotics to develop autonomous systems.

Fachkompetenzen: After successful completion of the module, students will be able to analyse and evaluate software components for autonomous mobile vehicles. The acquired competencies enable participants to apply and implement filtering techniques, particularly for self-localization and path planning, as well as to create nodes for the Robot Operating System (ROS2) and similar robotics frameworks.

Überfachliche Kompetenzen: Students acquire the ability to collaborate with robotics open-source communities and design and share new robotics applications using C++, Python, CMake and git. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Behaviour-Based Robotics
- Sensors Model (Laser Scanner) and Motion Model (Differential Drive)
- Mapping and Map Representation
- Self-localization (Discrete Filter, Particle Filter and Extended Kalman Filter)
- SLAM (Fast-SLAM, Kalman-based SLAM and Graph-based SLAM)
- Motion Planning and Path Planning
- Multi-Robot-Planning (ORCA, MStar, ...)

Erwartete Vorkenntnisse: Object-oriented programming skills and basic mathematical knowledge of statistics and probability theory.

These prerequisites are taught in the following modules:

- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Mobile Robotics

Model Engineering

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with model-driven approaches to software engineering. It combines techniques, methods, and tools from language engineering and model engineering.

Fachkompetenzen: Fundamental concepts and techniques of model-driven software engineering including in particular the development of domain-specific languages (metamodeling), concrete syntaxes, model transformators, and code generators; and the application thereof.

After successful completion of the module, students are able to

- apply model-driven software development or information system development to practical tasks,
- develop modeling languages and the required tool environment based on OMG's meta-modeling stack,
- evaluate transformation languages and use them for vertical, horizontal and temporal model transformation,
- evaluate and use textual and graphical modeling languages,
- evaluate language architectures, i.a. using the example of UML,
- use extension mechanisms of languages, i.a. UML profiles,
- use constraint languages, i.a. OCL to specify additional constraints on modeling languages,
- implement code generators, and
- solve tasks of model management, i.a. model evolution, model versioning and model storage.

Überfachliche Kompetenzen: Students acquire the ability to explain methods for model-driven software engineering and design domain-specific model-driven software engineering solutions. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Metamodeling
- Object Constraint Language (OCL)
- Textual Modeling Languages
- Graphical Modeling Languages
- Model Transformations
- ATLAS Transformation Language (ATL)
- Code Generation
- Guest Lectures

Erwartete Vorkenntnisse: Knowledge of "Software Engineering" and "Object-oriented modeling"

These prerequisites are taught in the following modules:

- Software Engineering (*Software Engineering*)
- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Model Engineering

Network Security

Regelarbeitsaufwand: 3,0–6,0 ECTS

Lernergebnisse:
Fachkompetenzen: After successful completion of *Network Security*, students are able to:

- explain basic concepts of network security
- describe how cryptographic methods work (including RSA, ElGamal, Diffie-Hellman, Elliptic Curve Cryptography)
- solve arithmetic problems on cryptographic methods
- describe and apply methods of anomaly detection
- use selected network data analysis and anomaly detection tools to analyze network data for different questions

After successful completion of *Network Security -Advanced Topics*, students are able to:

- describe selected concepts in network security (including BGP and MANET routing security, security concepts in IPv6 networks, in-depth cryptographic techniques) and apply them in examples
- solve arithmetic problems on cryptographic methods
- explain concepts of network steganography (covert channels, subliminal channels) and apply them in examples
- use selected network data analysis and anomaly detection tools to solve network steganography detection exercises

Überfachliche Kompetenzen: Students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt: *Network Security*: Security concepts and anomaly detection in network data: security objectives, threats and attack techniques, cryptography basic concepts and selected topics such as RSA, ElGamal, Diffie-Hellman and Elliptic Curve Cryptography, network traffic analysis and anomaly detection methods, practical use of network analysis tools.

Network Security - Advanced Topics: IPv6 security, routing security, secure group communication, network steganography (hidden communication in covert channels, subliminal channels) and future challenges (such as smart grid security), practical exercises on detecting network steganography in network traffic.

Erwartete Vorkenntnisse: Basic knowledge about communication networks, especially IP networks.

These prerequisites can be acquired in the bachelor module *Computersysteme*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
3,0/2,0 VU Network Security
3,0/2,0 VU Network Security - Advanced Topics

Problem Solving and Search in Artificial Intelligence

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: Students will learn about core AI problem solving techniques and their applications in various domains.

Fachkompetenzen: After successful completion of the module, students are able to:

- explain fundamental concepts of search methods in AI;
- model problems using solver-independent constraint modeling languages such as MiniZinc;
- explain and implement constraint programming techniques;
- use solvers based on constraint programming and SAT to solve various problems;
- explain structural decomposition techniques;
- explain and apply methods for automated algorithm selection and configuration;
- apply instance space analysis to evaluate the difficulty of problem instances and the impact of problem features;
- explain and implement local search and hyper-heuristic approaches; and
- apply learned techniques to solve complex practical problems, including those in planning and scheduling domains

Überfachliche Kompetenzen: Students acquire the ability to assess and use AI problem solving techniques in different application domains. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content. This includes the transparency and explainability of AI algorithms, and the societal impact of automated problem solving.

Inhalt: The module includes the following topics:

- solving problems by searching;
- constraint satisfaction/optimization problems;
- constraint programming techniques;
- solver-independent constraint modelling language MiniZinc;
- the structure of problems;
- local search techniques;
- application of machine learning in search;
- automated algorithm selection and configuration;

- hyper-heuristics; and
- instance space analysis
- AI-based scheduling and planning.

Erwartete Vorkenntnisse: Knowledge of programming and the basics of algorithms. These prerequisites are covered in the following modules:

- *Einführung in die Programmierung* and
- *Algorithmen und Datenstrukturen.*

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Problem Solving and Search in Artificial Intelligence

Processing of Declarative Knowledge

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:

Fachkompetenzen: After a successful completion of the module, students are be able to

- design and declaratively specify (possibly recursive) Datalog queries,
- develop ASP programs and queries, and classify such programs by their language features and computational properties,
- understand the principles of query optimization via magic set transformation,
- describe the structure and concepts of description logics (DLs), as well as distinguish and compare different DLs,
- identify the difference between (logic programming) rules and ontologies, and
- use formalisms for combining rules with other paradigms (e.g., ontologies) and languages.

Überfachliche Kompetenzen: Furthermore, students are able to

- differentiate between *declarative* and *procedural* problem solving approaches,
- extract accurate and complete problem specifications from natural language descriptions,
- develop and optimize solutions for different types of declarative specifications, and
- critically assess the suitability of different declarative solving techniques for different real-world problems of interest.

Moreover, the students are capable of identifying, verbalizing, and discussing the ethical considerations that arise when applying declarative knowledge processing techniques in real-world scenarios.

The material taught in this module does not have inherent gender aspects; the module focuses on foundational skills based on formal logic and mathematics. Teaching resources and methods will be developed with an awareness of diversity and ensuring a welcoming environment for students from underrepresented groups.

Inhalt: In this module, we consider different formalisms for representing and processing declarative knowledge, with particular emphasis on

- Datalog and extensions;
- optimization of Datalog queries;
- answer-set programming;
- description logics as languages for ontologies;
- computational aspects of reasoning;
- combining rules with other formalisms and languages (e.g., hybrid knowledge with rules and ontologies); and
- contextual Reasoning.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Processing of Declarative Knowledge

Program Analysis

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module explores foundational techniques in static program analysis, program semantics, and abstract interpretation, equipping students to reason about program behavior and apply these methods in areas such as verification, optimization, and software quality.

Fachkompetenzen: Students learn techniques and tools designed to analyze and understand the behavior of software programs. The module focuses on methods that allow for the inspection of software code without the need for execution, enabling the identification of potential bugs, security vulnerabilities, and performance bottlenecks.

Überfachliche Kompetenzen: This module equips students with the ability to apply formal methods and mathematical reasoning to analyze and improve software systems.

Inhalt:

- Static analyses for control flow, data flow, and dependence analysis;

- abstract interpretation and lattice-based reasoning for program properties;
- program semantics and its role in understanding program behavior;
- type systems and their role in ensuring program correctness;
- explore the practical applications of program analysis in optimization, verification, and software quality.

Erwartete Vorkenntnisse: Students should have a solid foundation in programming, algorithms, and data structures, as these are essential for understanding and applying the analysis techniques covered in the module. Familiarity with formal methods, logic, or discrete mathematics is beneficial.

These prerequisites are taught in the following modules:

- *Einführung in die Programmierung*;
- *Algorithmen und Datenstrukturen*;
- *Theoretische Informatik*; and
- *Algebra und Diskrete Mathematik*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Program Analysis

Programming Paradigms and Languages

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with programming paradigms and programming languages, especially relationships between paradigms, languages, and the ways we use them.

Fachkompetenzen:

- Describe major aspects of the design space of programming languages, goals and forces behind programming paradigms, strengths and pitfalls of paradigms as well as ways to deal with them.
- Develop small programs in several, partly unfamiliar languages of different paradigms, thereby selectively exploiting strengths and avoiding pitfalls.
- Implement a simple interpreter for a new tiny programming language.

Überfachliche Kompetenzen: Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Fundamental concepts of programming languages
- Basic language implementation techniques
- Language design decisions leading to different programming paradigms
- Goals, strengths and pitfalls of frequently used programming paradigms
- Language support for specific programming paradigms
- Differentiation between several kinds of abstraction in paradigms and languages
- Typical fields of application of paradigms and languages
- Combined use of several paradigms

Erwartete Vorkenntnisse: Knowledge of programming and major programming paradigms

Programming Paradigms (*Programmierparadigmen*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Programming Paradigms and Languages

Programming Principles of Mobile Robotics

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module covers state-of-the-art algorithms and methodologies from the domain of autonomous robotic systems like self-driving cars, drones, or search-and-rescue robots.

Fachkompetenzen: After successful completion of the module, students will be able to utilize, apply, and improve state-of-the-art algorithms, concepts, and techniques from the domain of autonomous mobile robotics for research and development.

Überfachliche Kompetenzen: Participants of this module will be able to utilize as much as actively work on state-of-the-art open-source robotic frameworks, to prototype own ideas, and develop full fledge robotics applications. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics within the context of the module's content.

Inhalt: In this module, principles, algorithms and methods for mobile autonomous and semi-autonomous robotic systems are taught. Selected problems with corresponding solutions from the fields of

- Software-Architecture and Frameworks
- Simulation
- Artificial Intelligence and Cognitive Robotics
- Cyber-physical-systems

are discussed as well as implemented by participants on domain specific hardware.

Erwartete Vorkenntnisse: Participation in this module requires good skills in software development and programming, as well as basic knowledge of mathematics and statistics.

These prerequisites can for example be acquired in following modules:

- Software Engineering (*Software Engineering*)
- Programming (*Einführung in die Programmierung*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Programming Principles of Mobile Robotics

Project in Computer Science

Regelarbeitsaufwand: 6,0–12,0 ECTS

Lernergebnisse:

Fachkompetenzen: Students can

- identify the background knowledge that is required for a task.
- apply the knowledge to solve the given task.
- justify the solution scientifically.

More concretely, they will

- apply scientific analysis, design, and implementation strategies (taking into account the state of the art).
- select suitable formal and mathematical methods for model building, abstraction, solution finding, and evaluation.
- use suitable technologies, software tools, and standards to solve the given task.
- thoroughly analyze the results and compare to other solutions proposed in the state of the art.
- document the results in a comprehensive and precise way.
- convincingly present the results in an interdisciplinary environment.

Überfachliche Kompetenzen: Students learn to

- formulate and solve problems independently, on their own and in groups.
- present problems and solutions.
- take into account critical comments on their work.
- judge their own limits and abilities.

The module also fosters individual creativity and innovation potential (curiosity). Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt: In this module, practical problems from the field of Computer Science are solved. This gives insights into scientific practice and current research in Computer Science. To tackle larger problems, two practical projects can be combined. Concretely, the work involves:

- Technical specification of a task from Computer Science.
- Scientifically sound solution of the given task.
- Scientific documentation and discussion of the results.

Erwartete Vorkenntnisse: The expected competencies and knowledge depend on the specific task.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls: Depending on the magnitude of the task, both certificates can also be awarded for completing a single, larger project.
6,0/4,0 PR Project in Computer Science 1
6,0/4,0 PR Project in Computer Science 2

Quantum Computing

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:

Fachkompetenzen: After the successful completion of the module, students are able to name and explain the most important methods in quantum computing, their central properties, and applications in different areas.

Überfachliche Kompetenzen: After successfully completing of the module, students are able to

- analyse employed techniques and methods,
- select relevant techniques and methods for a given problem, and
- critically assess relevant solutions and formalisms.

Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Overview;

- mathematical background and principles of quantum mechanics;
- quantum gates and quantum circuits;
- basic and advanced quantum algorithms;
- quantum complexity classes;
- elements of quantum information theory;
- quantum teleportation.

Erwartete Vorkenntnisse: Basic mathematical skills and knowledge of algorithmics. All prerequisites are covered in *Discrete Mathematics* and *Algorithmics*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Quantum Computing

Reinforcement Learning

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module provides an introduction into reinforcement learning, one of the main fields of machine learning.

Fachkompetenzen: After successful completion of the module, students are able to explain and to apply the theory and the methods of reinforcement learning, and also to implement the most fundamental algorithms.

Überfachliche Kompetenzen: Students acquire the ability to understand methods for machine learning and artificial intelligence and to solve problems in (stochastic) optimal control. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Data structures such as Markov decision processes;
- dynamic programming;
- Monte-Carlo algorithms;
- temporal-difference learning such as Q-learning;
- tabular methods vs. approximate solutions;
- on-policy vs. off-policy learning;
- eligibility traces;
- policy-gradient methods;
- convergence;
- deep reinforcement learning;
- applications such as solving chess, Go, shogi, Atari 2600 games, and other computer games.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Reinforcement Learning

Responsible Digital Ethics

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:
Fachkompetenzen: After successful completion of the course, students are able to:

- explain key concepts of digital ethics, including trust, fairness, accountability, and digital rights;
- recognize and apply regulatory frameworks and their implications for ethical digital practices;
- evaluate ethical design principles in the development and deployment of digital systems;
- analyze real-world digital ethics challenges using case studies and scholarly literature;
- critically evaluate ethical trade-offs in complex technological environments; and
- create actionable strategies to incorporate ethical principles into the design, audit, and deployment of digital systems.

Überfachliche Kompetenzen: Students are able to

- describe and reflect on personal and professional responsibilities as practitioners shaping the digital landscape,
- apply effective communication strategies about ethical challenges and solutions to diverse audiences, and
- design and implement frameworks for ethical practices in interdisciplinary and organizational contexts.

Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Introduction to digital ethics: the rise of digital systems and the imperative for ethical practices.
- Core principles: trust, fairness, accountability, transparency, and digital rights.
- Ethical design: principles and methodologies for designing ethical systems.
- Regulatory landscapes: overview of regulatory efforts and their impact on system performance and ethical audits.

- Practical challenges: case studies on ethical dilemmas in digital technology (e.g., AI bias, privacy concerns, algorithmic accountability).
- Roles and responsibilities: the critical role of practitioners in fostering responsible digital ethics.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Responsible Digital Ethics

SAT Algorithms, Applications and Extensions

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with the algorithms, and practical applications of SAT and SAT-based automated reasoning tools.

Fachkompetenzen: After successful completion of the module, students are able to understand fundamental concepts and techniques of modern SAT-based automated reasoning tools and their practical applications.

Überfachliche Kompetenzen: Students acquire the ability to understand the underlying methods of several automated decision procedures and learn how to adapt them to solve new problems.

Inhalt:

- Algorithms and advanced data structures of SAT solvers;
- formula simplification techniques in SAT solvers;
- techniques for certifying and checking SAT solver results;
- incremental SAT reasoning;
- introduction to the algorithm and underlying structure of CDCL(T) SMT solvers;
- different MaxSAT solving techniques, such as core-guided, model-improving, and hitting set based approaches;
- algorithm of search-based QBF solvers;
- modern applications of SAT-based reasoning tools.

Erwartete Vorkenntnisse: Attending students are assumed to be familiar with propositional and predicate logic, and have already some background in algorithms, formal methods, and practical software development.

These prerequisites are taught in the following modules:

- *Logic and Reasoning in Computer Science*;
- *Formal Methods in Systems Engineering*;

- *Algorithmen und Datenstrukturen*; and
- *Programmierparadigmen*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU SAT Algorithms, Applications and Extensions

Seminar in Computer Science

Regelarbeitsaufwand: min. 3,0 ECTS

Lernergebnisse:

Fachkompetenzen: Students are able to

- *find literature in a selected topic:* Students conduct independent literature research using academic databases, journals, conference proceedings, and other sources relevant to Computer Science. They critically evaluate sources for relevance, credibility, and scientific merit.
- *determine and apply an appropriate categorization for the found literature:* Students analyze the collected literature, identifying key themes, methodologies, and research directions. They develop a structured categorization scheme (taxonomies), grouping related works and highlighting connections and differences among them.
- *describe the covered research:* Students summarize and synthesize the findings from the literature, providing explanations of fundamental concepts, methodologies, and results. They discuss state-of-the-art approaches, challenges, and open research questions. Their descriptions include critical assessments of the strengths and limitations of various studies and their implications for the field of Computer Science.

Überfachliche Kompetenzen: Students can present their findings to a group in a structured and concise way. Students develop and deliver oral presentations that clearly communicate their literature analysis and key insights. They also demonstrate the ability to handle audience questions thoughtfully and participate in academic discussions. Additionally, students submit a written report on the content of their presentation, following academic writing standards and proper practices. The report provides a comprehensive overview of the chosen topic, reflecting deep understanding and critical analysis. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt: In the seminar, students dive deeper into the literature of a chosen topic in Computer Science and present their findings in written and oral form.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
3,0/2,0 SE Seminar in Computer Science

Smart Contracts

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:
Fachkompetenzen: Students acquire advanced programming skills specific to the blockchain paradigm. They develop a thorough understanding of the protocols, interfaces, and software infrastructure for developing decentralized systems. The course provides students with the analytical skills necessary to assess smart contracts for potential vulnerabilities and to apply secure development practices that mitigate the risk of exploitation. Furthermore, students are introduced to foundational concepts in program analysis, gaining insight into formal methods and automated techniques for the detection of software weaknesses. They are also trained in the effective use of contemporary analysis tools to support the development of robust and secure blockchain applications.
Überfachliche Kompetenzen: Students understand the interactions of decentralized applications with systems outside the blockchain. They develop awareness of the critical importance of software reliability and security in the blockchain context, where the remediation of errors is often impossible. The course further fosters students' capacity to design and implement secure applications through both theoretical grounding and practical training. Ethical considerations are integral to the curriculum, including a dedicated focus on responsible disclosure practices and the principles of ethical hacking as a means to strengthen the security and resilience of blockchain-based systems.

Inhalt: The module covers the following topics.

- programming languages and programming techniques for smart contracts;
- protocols, interfaces and technologies for blockchain applications;
- weaknesses and mitigation strategies in smart contracts; and
- methods and tools for analyzing smart contracts automatically.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Smart Contracts

Structural Decompositions and Meta Theorems

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with structural graph decompositions and their algorithmic applications through meta-theorems.

Fachkompetenzen: Students learn how such meta-theorems arise from the interaction between logic and structure, and how to apply them to problems in AI and database theory.

Überfachliche Kompetenzen: Students acquire the ability to combine logic theory, on the one hand, with algorithm design and analysis, on the other. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender and diversity in the context of structural decompositions and algorithmic meta-theorems.

Inhalt:

- Graph width parameters including tree-width and its generalizations;
- algorithmic meta-theorems and their applications;
- applications to AI and database theory, including propositional model counting, conjunctive query evaluation, and Bayesian Networks;
- well-behaved graph classes and their algorithmic implications.

Erwartete Vorkenntnisse: A solid understanding of algorithms, graph theory as well as fist-order logic is required.

These prerequisites are taught in the following modules:

- *Algorithmen und Datenstrukturen*;
- *Effiziente Algorithmen*;
- *Discrete Mathematics*; and
- *Algorithmics* (strongly recommended).

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Structural Decompositions and Meta Theorems

Symmetric Cryptography

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with attack and proof techniques for building symmetric cryptographic algorithms for secure data communication, storage, and computation.

Fachkompetenzen: Upon successful completion of the module, the students are able to:

- explain, argue the security of, and identify generic attacks on (tweakable) block ciphers, forkciphers, and compression functions in encryption, message authentication, hashing, authenticated encryption, and key derivation modes, among others;
- identify and reason about suitable use cases and applications of symmetric cryptographic schemes in higher level protocols and systems like TLS, IoT, end-to-end encryption, blockchains, messaging protocols, storage encryption, etc.;
- pin down lightweight symmetric cryptographic security requirements and design criteria;
- prove security of basic symmetric designs in the provable framework;
- design and attack implementation of small-scale toy symmetric designs;
- assess and compare existing symmetric designs with respect to security levels, efficiency, and applicability features; and
- present proofs and implementations, explain in detail contemporary use cases, and attacks and/or security proofs on symmetric algorithms.

Überfachliche Kompetenzen: Students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt: The module contains the following topics:

- provable security and notions in symmetric cryptography: pseudorandom functions and permutations, indistinguishability, unforgeability, collision resistance, etc.;
- designs and implementation strategies of building blocks: (tweakable) block ciphers, forkciphers, permutations, compression functions, etc.;
- secure modes of operation: encryption, message authentication, hashing, authenticated encryption, key derivation modes, etc.;
- symmetric lightweight cryptographic designs and the ongoing NIST competition;
- attacks and crypto failures;
- applications of symmetric cryptography: communication protocols like TLS, end-to-end encryption protocols, etc.

Erwartete Vorkenntnisse: Basic knowledge in security/cryptography and probability theory, basic programming skills.

These prerequisites are taught in the following modules:

- *Einführung in Security*;
- *Introduction to Cryptography*;
- *Statistik und Wahrscheinlichkeitstheorie*; and
- *Einführung in die Programmierung*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Symmetric Cryptography

System and Application Security

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse:
Fachkompetenzen: After successfully completing the module, students are able to:

- independently investigate security challenges in complex systems
- develop necessary tooling to help identify and verifying vulnerabilities
- communicate solutions and best practices to prevent security problems

Überfachliche Kompetenzen: Students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Security of Mobile Systems
- Software Security
- Secure Development Best Practices
- Web Security
- Security of Networked Systems
- Reverse Engineering
- Physical aspects of Security (e.g. side-channels)
- Discovery of Vulnerabilities
- Malware Ecosystems
- Risk and Threat Modelling

Erwartete Vorkenntnisse: Students are expected to know fundamentals of security such as Confidentiality, Integrity, Availability, Access Control, Web Security, and fundamentals of cryptography such as Hashing, Symmetric, and Asymmetric Cryptography.

We also expect familiarity of security aspects of operating systems, mobile systems, and networking technologies.

This module assumes the previous knowledge conveyed in the bachelor modules *Einführung in Security* and *Foundations of System and Application Security*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU System and Application Security

Theoretical Foundations and Research Topics in Machine Learning

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with the algorithmic and theoretical foundations of machine learning algorithms, as well as with recent advancements in the machine learning literature.

Fachkompetenzen: After successful completion of the module, students are able to:

- explain the theoretical foundations of machine learning;
- prove learning theoretical results and algorithmic properties of machine learning;
- apply learning algorithms correctly;
- compare and analyse learning algorithms; and
- explain, summarise, and present machine learning research papers.

Überfachliche Kompetenzen: Students acquire the ability to understand and apply machine learning in new application domains. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Empirical risk minimisation and regularisation;
- probably approximately correct (PAC) learning;
- Vapnik–Chervonenkis (VC) dimension;
- kernel-based learning and support vector machines;
- least squares regression;
- deep learning and graph neural network (GNN).

Erwartete Vorkenntnisse: Prior programming skills and fundamental maths skills are expected.

These prerequisites are covered in the module *Machine Learning*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Theoretical Foundations and Research Topics in Machine Learning

Theoretical Foundations of Deep Learning

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with the theoretical foundations of deep learning.

Fachkompetenzen: After successful completion of the module, students are able to:

- explain various deep network architectures, including the scattering transform, and key components such as nonlinear transforms, pooling, convolutional structures, and training methods; and
- explain theoretical insights into the performance of deep networks, including topics like dictionary learning, early-layer transferability, energy decay with depth, Lipschitz continuity, depth's role in overcoming the curse of dimensionality, adversarial example construction, network geometry via random matrix theory, and invariance learning.

Überfachliche Kompetenzen: Students acquire foundational insights into deep learning technologies and develop the ability to use these insights to assess both the use-cases and limitations of deep learning across different application domains. Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt: The module contains the following topics:

- ingredients of deep learning,
- deep learning: application and approximation;
- exponential expressivity with depth;
- DNNs can overcome the curse of dimensionality;
- exponential growth of variance and correlation;
- variance of the Jacobian's spectrum;
- stochastic gradient descent and its extensions;
- optimization algorithms for training DNNs;
- topology of the loss landscape;
- loss landscape: impact of parameterization and architecture;
- visualising the filters in a CNN;
- the scattering transform;
- autoencoders;
- GANs and adversarial examples; and
- physics-informed neural networks.

Erwartete Vorkenntnisse: This module requires only basic knowledge of linear algebra and probability.

These prerequisites are taught in the module *Machine Learning*.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Theoretical Foundations of Deep Learning

Theory of Graph Data

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: The goal of this module is to provide a solid understanding of the foundations of the technologies for graph-structured data in order to compare their uses, limits and possibilities. Studying them through the lens of database theory, computational logic, and computational complexity, the students will achieve an understanding that abstracts from the concrete data models and systems, enabling the students to effectively leverage the potential of existing technologies and assess their limitations.

Fachkompetenzen: On successful completion of the module, students will be able to

- compare and critique modern technologies for graphically structured data,
- apply finite state automata methods to various aspects of graph-structured data,
- translate modelling and constraint languages for graph-structured data into description logics (DLs),
- evaluate the computational cost of different features offered by graph database technologies,
- compare different available technologies based on their expressiveness and computational cost, and
- assess the capabilities and limitations of new and emerging graph database technologies.

Überfachliche Kompetenzen: Furthermore, students are able to contrast different formalisms, establish relationships between them and assess their appropriateness for different problems. They will be able to identify common features of different types of formalisms and to abstract mathematical formulations that unify different languages. The material taught in this module does not have inherent gender or ethical aspects; the module focuses on foundational skills based on formal logic and mathematics. Teaching resources and methods will be developed with an awareness of diversity and ensuring a welcoming environment for students from underrepresented groups.

Inhalt:

- Various models of finite state automata (deterministic, non-deterministic, alternating), including automata over words and automata over trees; conversions between selected models of automata
- regular expressions and their connection to automata;
- various models for graph-structured data;
- (labeled) trees as a special case of graph-structured data;
- navigational query languages for graph-structured data;
- notions of data and combined complexity for query languages;
- expressiveness and computational complexity of various features of query languages for graph-structured data;
- description logics (DLs) as languages for modeling graph-structured data;
- DLs as languages for integrity constraints over graph-structured data;

- basic algorithms and complexity results for reasoning tasks related to modelling graph-structured using DLs;
- basic algorithms and complexity results for constraint languages based on DLs;
- graph-structured data as (non-materialized) views over relational data sources;
- selected topics on recent advances in the area.

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Theory of Graph Data

Type Systems

Regelarbeitsaufwand: 6,0 ECTS

Lernergebnisse: This module deals with type systems in programming languages including their formal foundations and practical use in functional and object-oriented programming.

Fachkompetenzen:

- Describe different kinds of type systems, their potentials and limits, type checking methods as well as typical applications using technical vocabulary.
- Use types to specify abstractions with given properties.
- Implement a simple type checker for a simple programming language.

Überfachliche Kompetenzen: Furthermore, students will be able to identify, articulate, and discuss issues concerning ethics, gender, and diversity in the context of the module's content.

Inhalt:

- Meaning of the notions of type systems and types for programming languages
- Untyped λ-calculus versus simple typed λ-calculus and Y combinator
- Connections between simple logic theorem proving and type checking
- Algebra as foundation for specifying abstract data types
- Practical aspects of types in imperative programming
- Type inference in simple functional programming languages
- Practical aspects of types in functional programming
- Subtyping based on the principle of substitutability
- Higher-order subtyping
- Combining genericity and subtyping
- Practical aspects of types and abstract data types in object-oriented programming
- Behavioral subtyping and Liskov substitutability

- Propagating arbitrary static (type) information within a program
- Closer look on a more advanded type system as a showcase

Erwartete Vorkenntnisse:

- Knowledge of programming in a strongly typed language (e.g., Java)
- Knowledge of fundamental properties of basic programming paradigms

- Programming (*Einführung in die Programmierung*)
- Programming Paradigms (*Programmierparadigmen*)

Angewendete Lehr- und Lernformen und geeignete Leistungsbeurteilung: Die angewendeten Lehr- und Lernformen sind im Informationssystem zu Studien und Lehre bei jeder Lehrveranstaltung vor Beginn des Semesters anzugeben; ebenso die Prüfungsmodalitäten.

Lehrveranstaltungen des Moduls:
6,0/4,0 VU Type Systems

B Übergangsbestimmungen

1. Sofern nicht anders angegeben, wird im Folgenden unter Studium das *Masterstudium Software Engineering (Studienkennzahl UE 066 937)* verstanden. Der Begriff neuer Studienplan bezeichnet diesen ab 1.10.2025 für dieses Studium an der Technischen Universität Wien gültigen Studienplan und alter Studienplan den bis dahin gültigen. Entsprechend sind unter neuen bzw. alten Lehrveranstaltungen solche des neuen bzw. alten Studienplans zu verstehen (alt inkludiert auch frühere Studienpläne). Mit Studienrechtlichem Organ ist das für das Masterstudium Software Engineering zuständige Studienrechtliche Organ an der Technischen Universität Wien gemeint.

2. Die Übergangsbestimmungen gelten für Studierende, die den Studienabschluss gemäß neuem Studienplan an der Technischen Universität Wien einreichen und die vor dem 1.7.2025 zum Masterstudium Software Engineering an der Technischen Universität Wien zugelassen waren. Das Ausmaß der Nutzung der Übergangsbestimmungen ist diesen Studierenden freigestellt.

3. Auf Antrag der_des Studierenden kann das Studienrechtliche Organ die Übergangsbestimmungen individuell modifizieren oder auf nicht von Absatz 2 erfasste Studierende ausdehnen.

4. Zeugnisse über Lehrveranstaltungen, die inhaltlich äquivalent sind, können nicht gleichzeitig für den Studienabschluss eingereicht werden. Im Zweifelsfall entscheidet das Studienrechtliche Organ über die Äquivalenz.

5. Zeugnisse über alte Lehrveranstaltungen können, soferne im Folgenden nicht anders bestimmt, jedenfalls für den Studienabschluss verwendet werden, wenn die Lehrveranstaltung von der_dem Studierenden mit Stoffsemester Sommersemester 2025 oder früher absolviert wurde.

6. Überschüssige ECTS-Punkte aus den Pflichtmodulen können als Ersatz für zu erbringende Leistungen in Wahlmodulen sowie als Freie Wahlfächer und/oder Transferable Skills verwendet werden. Überschüssige ECTS-Punkte aus den Wahlmodulen können als Ersatz für zu erbringende Leistungen in den Freien Wahlfächern und/oder Transferable Skills verwendet werden.

7. Fehlen nach Anwendung der Bestimmungen aus den Äquivalenzlisten ECTS-Punkte zur Erreichung der notwendigen 120 ECTS-Punkte für den Abschluss des Masterstudiums, so können diese durch noch nicht verwendete Lehrveranstaltungen aus den Wahlmodulen und/oder Freien Wahlfächern und Transferable Skills im notwendigen Ausmaß abgedeckt werden.

8. Im Folgenden wird jede Lehrveranstaltung (*alt* oder *neu*) durch ihren Umfang in ECTS-Punkten (erste Zahl) und Semesterstunden (zweite Zahl), ihren Typ und ihren Titel beschrieben. Es zählt der ECTS-Umfang der tatsächlich absolvierten Lehrveranstaltung.

Die Lehrveranstaltungen auf der linken Seite der nachfolgenden Tabellen bezeichnen die alten Lehrveranstaltungen. Auf der rechten Seite sind die Lehrveranstaltungen angegeben, für welche die alten Lehrveranstaltungen jeweils verwendet werden können. Lehrveranstaltungen, die unter demselben Punkt in den Äquivalenzlisten angeführt sind, gelten als äquivalent.

Alt	Neu
3,0/2,0 VO Advanced Software Engineering	6,0/4,0 VU Advanced Software Engineering
3,0/2,0 VU Advanced Software Engineering	6,0/4,0 VU Advanced Software Engineering
3,0/2,0 VU Advanced Internet Computing	6,0/4,0 VU Advanced Internet Computing
6,0/4,0 VU Formal Methods in Computer Science	6,0/4,0 VU Formal Methods in Systems Engineering
4,5/3,0 VU Algorithmic Geometry und/oder 1,5/1,0 UE Algorithmic Geometry	6,0/4,0 VU Algorithmic Geometry
4,5/3,0 VU Algorithms in Graph Theory	6,0/4,0 VU Algorithms in Graph Theory
3,0/2,0 VU Approximation Algorithms	6,0/4,0 VU Beyond Exact Algorithms
4,5/3,0 VU Fixed-Parameter Algorithms and Complexity	6,0/4,0 VU Fixed-Parameter Algorithms and Complexity
4,5/3,0 VU Graph Drawing Algorithms	6,0/4,0 VU Graph Drawing Algorithms
4,5/3,0 VU Heuristic Optimization Techniques	6,0/4,0 VU Heuristic Optimization Techniques
4,5/3,0 VU Machine Learning	6,0/4,0 VU Machine Learning
3,0/2,0 VU Mathematical Programming und/oder 3,0/2,0 VU Optimization in Transport and Logistics	6,0/4,0 VU Mathematical Programming and Optimization in Transport Logistics
3,0/2,0 VU Problem Solving and Search in Artificial Intelligence	6,0/4,0 VU Problem Solving and Search in AI
3,0/2,0 VU Randomized Algorithms	6,0/4,0 VU Beyond Exact Algorithms
3,0/2,0 VU Structural Decompositions and Algorithms	6,0/4,0 VU Structural Decompositions and Meta Theorems
3,0/2,0 VU Effiziente Programme	3,0/2,0 VU Efficient Programs
3,0/2,0 VU Fortgeschrittene logische Programmierung	6,0/4,0 VU Advanced Logic Programming
3,0/2,0 VU Fortgeschrittene objektorientierte Programmierung und/oder 3,0/2,0 VU Programmiersprachen	6,0/4,0 VU Programming Paradigms and Languages

... Fortsetzung

Alt	Neu
6,0/4,0 VU GPU Architectures and Computing	6,0/4,0 VU GPU Computing and Architectures
4,5/3,0 VU High Performance Computing	6,0/4,0 VU High Performance Computing
3,0/2,0 VO Typsysteme	6,0/4,0 VU Type Systems
4,5/3,0 VU Weiterführende Multiprocessor Programmierung	6,0/4,0 VU Advanced Multiprocessor Programming
4,5/3,0 VU Mobile Robotics	6,0/4,0 VU Mobile Robotics
4,5/3,0 VU Hybrid Classic/Quantum Systems	6,0/4,0 VU Hybrid Classic-Quantum Systems
3,0/2,0 VU Computer-Aided Verification und/oder 3,0/2,0 UE Computer-Aided Verification	6,0/4,0 VU Computer-Aided Verification
3,0/2,0 VU Complexity Theory	6,0/4,0 VU Complexity Theory
3,0/2,0 VU Programmanalyse	6,0/4,0 VU Program Analysis
3,0/2,0 VU Quantum Computing	6,0/4,0 VU Quantum Computing
3,0/2,0 VU SAT Solving and Extensions	6,0/4,0 VU SAT Algorithms, Applications and Extensions
6,0/4,0 VU Business Intelligence	6,0/4,0 VU Business Intelligence
3,0/2,0 VU Datenbanktheorie	6,0/4,0 VU Database Theory
6,0/4,0 VU Datenbanksysteme Vertiefung	6,0/4,0 VU Advanced Database Systems
3,0/2,0 VU Deduktive Datenbanken und/oder 3,0/2,0 VU Processing of Declarative Knowledge	6,0/4,0 VU Processing of Declarative Knowledge
3,0/2,0 VU Einführung in Semantic Systems	6,0/4,0 VU Management of Graph Data
3,0/2,0 VU Knowledge Graphs	6,0/4,0 VU Knowledge Graphs

9. Die folgende Tabelle ordnet alte Lehrveranstaltungen ohne Entsprechung im neuen Studienplan einem der neuen Module zu. Lehrveranstaltungen mit Typ SE sind dem Modul *Seminar in Computer Science* zuzuordnen. Andere Lehrveranstaltung von Wahlmodulen einer früheren Studienplanversion die in der Tabelle nicht aufgelistet sind, sind dem Modul *Extension* zuzuordnen. Diese Lehrveranstaltungen zählen jedoch nicht zum ECTS-Limit des Moduls *Extension*.

Alt	Neu
6,0/4,0 VU Advanced Algorithms	Advanced Topics In Algorithms and Complexity
4,5/3,0 VU Dependable Distributed Systems	Advanced Topics In Algorithms and Complexity

... Fortsetzung

Alt	Neu
3,0/2,0 VU Discrete Reasoning Methods	Advanced Topics In Algorithms and Complexity
6,0/4,0 VU Distributed Algorithms	Advanced Topics In Algorithms and Complexity
3,0/2,0 VU Efficient Algorithms	Advanced Topics In Algorithms and Complexity
3,0/2,0 VU Algorithms Design	Advanced Topics In Algorithms and Complexity
5,0/3,0 VO Analysis of Algorithms	Advanced Topics In Algorithms and Complexity
4,0/2,0 UE Analysis of Algorithms	Advanced Topics In Algorithms and Complexity
3,0/2,0 VO Inductive Rule Learning	Advanced Topics In Algorithms and Complexity
3,0/2,0 VU Modeling and Solving Constrained Optimization Problems	Advanced Topics In Algorithms and Complexity
3,0/2,0 VU Networks: Design and Analysis	Advanced Topics In Algorithms and Complexity
4,5/3,0 VU Problems in Distributed Computing	Advanced Topics In Algorithms and Complexity
3,0/2,0 VU Real-Time Scheduling	Advanced Topics In Algorithms and Complexity
3,0/2,0 VU Parallele Algorithmen	Advanced Topics In Algorithms and Complexity
3,0/2,0 VU Advanced Project Management	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Advanced Model Engineering	Advanced Topics In Software Engineering and Programming
3,0/2,0 SE Advanced Model Engineering	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Configuration Management	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Management von Software Projekten	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Methods of Empirical Software Engineering	Advanced Topics In Software Engineering and Programming
6,0/4,0 VU Model Engineering	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Requirements Engineering and Specification	Advanced Topics In Software Engineering and Programming

... Fortsetzung

Alt	Neu
3,0/2,0 VU Risk Management	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Software Maintenance and Evolution	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Software Quality Management	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Software Testing	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Value-Based Software Engineering	Advanced Topics In Software Engineering and Programming
3,0/2,0 UE Formale Methoden der Informatik	Advanced Topics In Software Engineering and Programming
3,0/2,0 VO Analyse und Verifikation	Advanced Topics In Software Engineering and Programming
3,0/2,0 VO Codegeneratoren	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Dynamic Compilation	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Fortgeschrittene funktionale Programmierung	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Optimierende Übersetzer	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Stackbasierte Sprachen	Advanced Topics In Software Engineering and Programming
3,0/2,0 VU Übersetzer für Parallele Systeme	Advanced Topics In Software Engineering and Programming
6,0/4,0 VU Parallele und Echtzeitprogrammierung	Advanced Topics In Software Engineering and Programming
6,0/4,0 VU Programming Principles of Mobile Robotics	Advanced Topics In Software Engineering and Programming
3,0/2,0 VO Deduktive Datenbanken	Advanced Topics In Data Management and Intelligent Systems
3,0/2,0 VU Processing of Declarative Knowledge	Advanced Topics In Data Management and Intelligent Systems
3,0/2,0 VU Serverless Computing	Advanced Topics In Distributed and Next Generation Computing
6,0/4,0 VU Advanced Distributed Systems	Advanced Topics In Distributed and Next Generation Computing
4,0/3,0 VU Computer Networks	Advanced Topics In Distributed and Next Generation Computing

... Fortsetzung

Alt	Neu
3,0/2,0 VU Distributed Systems Engineering	Advanced Topics In Distributed and Next Generation Computing
6,0/4,0 VU Large-scale Distributed Computing	Advanced Topics In Distributed and Next Generation Computing
3,0/2,0 VU Mobile Network Services and Applications	Advanced Topics In Distributed and Next Generation Computing
2,0/2,0 VO Network Engineering	Advanced Topics In Distributed and Next Generation Computing
2,0/1,0 UE Network Engineering	Advanced Topics In Distributed and Next Generation Computing
3,0/2,0 VU Peer-to-Peer Systems	Advanced Topics In Distributed and Next Generation Computing
2,0/2,0 VO Pervasive and Mobile Computing	Advanced Topics In Distributed and Next Generation Computing
3,0/2,0 VU Software Architecture	Advanced Topics In Distributed and Next Generation Computing
3,0/2,0 VU Service Level Agreements	Advanced Topics In Distributed and Next Generation Computing
3,0/2,0 VU Software in Kommunikationsnetzen	Advanced Topics In Distributed and Next Generation Computing
6,0/4,0 VU Verteiltes Programmieren mit Space Based Computing Middleware	Advanced Topics In Distributed and Next Generation Computing
3,0/2,0 VU Web Application Engineering and Content Management	Advanced Topics In Distributed and Next Generation Computing
3,0/3,0 VU Wireless in Automation	Advanced Topics In Distributed and Next Generation Computing
3,0/2,0 VU Rigorous Systems Engineering	Advanced Topics In Verification and Automated Reasoning
4,5/3,0 VU Automated Reasoning and Program Verification	Advanced Topics In Verification and Automated Reasoning
3,0/2,0 VU Formal Methods for Concurrent and Distributed Systems	Advanced Topics In Verification and Automated Reasoning
3,0/2,0 VU Introduction to Type Theories	Advanced Topics In Verification and Automated Reasoning
6,0/4,0 VU Software Model Checking	Advanced Topics In Verification and Automated Reasoning
4,5/3,0 VU Semantik von Programmiersprachen	Advanced Topics In Verification and Automated Reasoning
3,0/2,0 VU Coalgebra in Computer Science	Advanced Topics In Verification and Automated Reasoning

... Fortsetzung

Alt	Neu
3,0/2,0 VU Advanced Security for Systems Engineering	Advanced Topics In Security and Privacy
3,0/2,0 VU Digital Forensics	Advanced Topics In Security and Privacy
3,0/2,0 VU IT security in Large IT infrastructures	Advanced Topics In Security and Privacy
3,0/2,0 VU Kryptographie	Advanced Topics In Security and Privacy

10. In den Prüfungsfächern *Algorithms and Complexity*, *Automation Systems and Mobile Robotics*, *Data Management and Intelligent Systems* und *Machine Learning* können im Rahmen dieser Übergangsbestimmungen Wahlmodule gewählt werden, ohne dass das entsprechende Core-Modul gewählt wird.

11. Studierende können im Rahmen dieser Übergangsbestimmungen ihr Studium mit der ursprünglichen Bezeichnung „Software Engineering & Internet Computing“ abschließen, wenn sie dies beim Einreichen des Studienabschlusses dem Studienrechtlichen Organ durch eine schriftliche Erklärung bekanntgeben.

12. Im Rahmen dieser Übergangsbestimmungen ist im Modul *Efficient Programs* die Lehrveranstaltung 3,0/2,0 VU Efficient Programs optional. Im Modul *Data Stewardship* sind die Lehrveranstaltung 3,0/2,0 VO Data Stewardship oder die Lehrveranstaltung 3,0/2,0 UE Data Stewardship oder beide zu absolvieren.

C Semestereinteilung der Lehrveranstaltungen

Es wird empfohlen, die Lehrveranstaltungen der Pflichtmodule in folgender Reihenfolge zu absolvieren.

1. Semester (WS)

6,0 VU Advanced Software Engineering
6,0 PR Advanced Software Engineering

2. oder 3. Semester

3,0 SE Seminar in Computer Science

D Prüfungsfächer mit den zugeordneten Modulen und Lehrveranstaltungen

Die mit ∗ markierten Module sind Wahlmodule, die mit + markierten Module sind Core-Module, die übrigen sind Pflichtmodule.

Prüfungsfach „Algorithms and Complexity“

+Modul „Algorithmics“ (6,0 ECTS)

6,0/4,0 VU Algorithmics

∗Modul „Advanced Research in Algorithmics“ (6,0 ECTS)

6,0/4,0 VU Advanced Research in Algorithmics

∗Modul „Algorithmic Encoding Techniques“ (6,0 ECTS)

6,0/4,0 VU Algorithmic Encoding Techniques

∗Modul „Algorithmic Geometry“ (6,0 ECTS)

6,0/4,0 VU Algorithmic Geometry

∗Modul „Algorithmic Social Choice“ (6,0 ECTS)

6,0/4,0 VU Algorithmic Social Choice

∗Modul „Algorithms in Graph Theory“ (6,0 ECTS)

6,0/4,0 VU Algorithms in Graph Theory

∗Modul „Beyond Exact Algorithms“ (6,0 ECTS)

6,0/4,0 VU Beyond Exact Algorithms

∗Modul „Complexity Theory“ (6,0 ECTS)

6,0/4,0 VU Complexity Theory

∗Modul „Fixed-Parameter Algorithms and Complexity“ (6,0 ECTS)

6,0/4,0 VU Fixed-Parameter Algorithms and Complexity

∗Modul „Graph Drawing Algorithms“ (6,0 ECTS)

6,0/4,0 VU Graph Drawing Algorithms

∗Modul „Heuristic Optimization Techniques“ (6,0 ECTS)

6,0/4,0 VU Heuristic Optimization Techniques

∗Modul „Mathematical Programming and Optimization in Transport Logistics“ (6,0 ECTS)

6,0/4,0 VU Mathematical Programming and Optimization in Transport Logistics

∗Modul „Structural Decompositions and Meta Theorems“ (6,0 ECTS)

6,0/4,0 VU Structural Decompositions and Meta Theorems

∗Modul „Advanced Topics In Algorithms and Complexity“ (min. 3,0 ECTS)

Prüfungsfach „Automation Systems and Mobile Robotics“

+Modul „Mobile Robotics“ (6,0 ECTS)

6,0/4,0 VU Mobile Robotics

∗Modul „Autonomous Racing Cars“ (6,0 ECTS)

6,0/4,0 VU Autonomous Racing Cars

∗Modul „Information Technology in Automation“ (6,0 ECTS)

6,0/4,0 VU Information Technology in Automation

∗Modul „Programming Principles of Mobile Robotics“ (6,0 ECTS)

6,0/4,0 VU Programming Principles of Mobile Robotics

∗Modul „Advanced Topics In Automation and Mobile Robotics“ (min. 3,0 ECTS)

Prüfungsfach „Data Management and Intelligent Systems“

+Modul „Advanced Database Systems“ (6,0 ECTS)

6,0/4,0 VU Advanced Database Systems

∗Modul „Business Intelligence“ (6,0 ECTS)

6,0/4,0 VU Business Intelligence

∗Modul „Database Theory“ (6,0 ECTS)

6,0/4,0 VU Database Theory

∗Modul „Data Stewardship“ (6,0 ECTS)

3,0/2,0 VO Data Stewardship
3,0/2,0 UE Data Stewardship

∗Modul „Knowledge Graphs“ (6,0 ECTS)

6,0/4,0 VU Knowledge Graphs

∗Modul „Logic-based Artificial Intelligence“ (6,0 ECTS)

6,0/4,0 VU Logic-based Artificial Intelligence

∗Modul „Management of Graph Data“ (6,0 ECTS)

6,0/4,0 VU Management of Graph Data

∗Modul „Problem Solving and Search in Artificial Intelligence“ (6,0 ECTS)

6,0/4,0 VU Problem Solving and Search in Artificial Intelligence

∗Modul „Processing of Declarative Knowledge“ (6,0 ECTS)

6,0/4,0 VU Processing of Declarative Knowledge

∗Modul „Theory of Graph Data“ (6,0 ECTS)

6,0/4,0 VU Theory of Graph Data

∗Modul „Advanced Topics In Data Management and Intelligent Systems“ (min. 3,0 ECTS)

Prüfungsfach „Distributed and Next Generation Computing“

+Modul „Advanced Internet Computing“ (6,0 ECTS)

6,0/4,0 VU Advanced Internet Computing

+Modul „Distributed Systems Technologies“ (6,0 ECTS)

6,0/4,0 VU Distributed Systems Technologies

∗Modul „Artifact-based Design“ (6,0 ECTS)

6,0/4,0 VU Artifact-based Design

∗Modul „Hybrid Quantum - Classical Systems“ (6,0 ECTS)

6,0/4,0 VU Hybrid Quantum - Classical Systems

∗Modul „Internet of Things“ (6,0 ECTS)

6,0/4,0 VU Internet of Things

∗Modul „Quantum Computing“ (6,0 ECTS)

6,0/4,0 VU Quantum Computing

∗Modul „Advanced Topics In Distributed and Next Generation Computing“ (min. 3,0 ECTS)

Prüfungsfach „High Performance Computing“

∗Modul „Advanced Multiprocessor Programming“ (6,0 ECTS)

6,0/4,0 VU Advanced Multiprocessor Programming

∗Modul „GPU Computing and Architectures“ (6,0 ECTS)

6,0/4,0 VU GPU Computing And Architectures

∗Modul „Green HPC“ (6,0 ECTS)

6,0/4,0 VU Green HPC

∗Modul „High Performance Computing“ (6,0 ECTS)

6,0/4,0 VU High Performance Computing

∗Modul „HPC for AI“ (6,0 ECTS)

6,0/4,0 VU HPC for AI

∗Modul „Advanced Topics In High Performance Computing“ (min. 3,0 ECTS)

Prüfungsfach „Machine Learning“

+Modul „Machine Learning“ (6,0 ECTS)

6,0/4,0 VU Machine Learning

∗Modul „Advanced Reinforcement Learning“ (6,0 ECTS)

6,0/4,0 VU Advanced Reinforcement Learning

∗Modul „Algorithms for Data Science“ (6,0 ECTS)

6,0/4,0 VU Algorithms for Data Science

∗Modul „Applied Generative AI and LLM-based Systems“ (6,0 ECTS)

6,0/4,0 VU Applied Generative AI and LLM-based Systems

∗Modul „Deep Learning for Natural Language Processing“ (6,0 ECTS)

6,0/4,0 VU Deep Learning for Natural Language Processing

∗Modul „Generative AI“ (6,0 ECTS)

6,0/4,0 VU Generative AI

∗Modul „Machine Learning for Optimization“ (6,0 ECTS)

6,0/4,0 VU Machine Learning for Optimization

∗Modul „Reinforcement Learning“ (6,0 ECTS)

6,0/4,0 VU Reinforcement Learning

∗Modul „Theoretical Foundations and Research Topics in Machine Learning“ (6,0 ECTS)

6,0/4,0 VU Theoretical Foundations and Research Topics in Machine Learning

∗Modul „Theoretical Foundations of Deep Learning“ (6,0 ECTS)

6,0/4,0 VU Theoretical Foundations of Deep Learning

∗Modul „Advanced Topics In Machine Learning“ (min. 3,0 ECTS)

Prüfungsfach „Security and Privacy“

∗Modul „Advanced Cryptography“ (6,0 ECTS)

6,0/4,0 VU Advanced Cryptography

∗Modul „Advanced Privacy Enhancing Technologies“ (6,0 ECTS)

6,0/4,0 VU Advanced Privacy Enhancing Technologies

∗Modul „Artificial Intelligence for Computer Security“ (6,0 ECTS)

6,0/4,0 VU Machine Learning for Computer Security

∗Modul „Cryptocurrencies“ (6,0 ECTS)

6,0/4,0 VU Cryptocurrencies

∗Modul „Formal Methods for Security and Privacy“ (6,0 ECTS)

6,0/4,0 VU Formal Methods for Security and Privacy

∗Modul „Network Security“ (3,0–6,0 ECTS)

3,0/2,0 VU Network Security
3,0/2,0 VU Network Security - Advanced Topics

∗Modul „Smart Contracts“ (6,0 ECTS)

6,0/4,0 VU Smart Contracts

∗Modul „Symmetric Cryptography“ (6,0 ECTS)

6,0/4,0 VU Symmetric Cryptography

∗Modul „System and Application Security“ (6,0 ECTS)

6,0/4,0 VU System and Application Security

∗Modul „Advanced Topics In Security and Privacy“ (min. 3,0 ECTS)

Prüfungsfach „Societal Impact and Critical Reflections“

∗Modul „Advanced Human-Centered AI: from concepts to implementation“ (6,0 ECTS)

6,0/4,0 VU Advanced Human-Centered AI: from concepts to implementation

∗Modul „AI Ethics“ (6,0 ECTS)

6,0/4,0 VU AI Ethics

∗Modul „Critical Algorithm Studies“ (6,0 ECTS)

6,0/4,0 VU Critical Algorithm Studies

∗Modul „Computer Science Education: Advances in Research and Practice “ (6,0 ECTS)

6,0/4,0 PR Computer Science Education: Advances in Research and Practice

∗Modul „Critical Theory of Media and Informatics“ (6,0 ECTS)

6,0/4,0 VU Critical Theory of Media and Informatics

∗Modul „Human-agent Interaction“ (12,0 ECTS)

6/4,0 VU Human-agent Interaction
6,0/4,0 PR Human-agent Interaction

∗Modul „Introduction to Computational Sustainability“ (6,0 ECTS)

6,0/4,0 VU Introduction to Computational Sustainability

∗Modul „Learning Technologies and Learning Analytics“ (6,0 ECTS)

6,0/4,0 VU Learning Technologies and Learning Analytics

∗Modul „Responsible Digital Ethics“ (6,0 ECTS)

6,0/4,0 VU Responsible Digital Ethics

∗Modul „Advanced Topics In Societal Impact and Critical Reflections“ (min. 3,0 ECTS)

Prüfungsfach „Software Engineering and Programming“

Modul „Advanced Software Engineering“ (6,0 ECTS)

6,0/4,0 VU Advanced Software Engineering

Modul „Advanced Software Engineering Project“ (6,0 ECTS)

6,0/4,0 PR Advanced Software Engineering

∗Modul „Advanced Logic Programming“ (6,0 ECTS)

6,0/4,0 VU Advanced Logic Programming

∗Modul „Advanced Model Engineering“ (6,0 ECTS)

6,0/4,0 VU Advanced Model Engineering

∗Modul „AI Programming“ (6,0 ECTS)

6,0/4,0 VU AI Programming

∗Modul „Efficient Programs“ (6,0 ECTS)

3,0/2,0 VU Efficient Programs
3,0/2,0 PR Efficient Programs

∗Modul „Low-Level Programming“ (6,0 ECTS)

6,0/4,0 VU Low-Level Programming

∗Modul „Model Engineering“ (6,0 ECTS)

6,0/4,0 VU Model Engineering

∗Modul „Programming Paradigms and Languages“ (6,0 ECTS)

6,0/4,0 VU Programming Paradigms and Languages

∗Modul „Type Systems“ (6,0 ECTS)

6,0/4,0 VU Type Systems

∗Modul „Advanced Topics In Software Engineering and Programming“ (min. 3,0 ECTS)

Prüfungsfach „Verification and Automated Reasoning“

+Modul „Formal Methods in Systems Engineering“ (6,0 ECTS)

6,0/4,0 VU Formal Methods in Systems Engineering

∗Modul „Automata and Logic“ (6,0 ECTS)

6,0/4,0 VU Automata and Logic

∗Modul „Automated Deduction“ (6,0 ECTS)

6,0/4,0 VU Automated Deduction

∗Modul „Computer-Aided Verification“ (6,0 ECTS)

6,0/4,0 VU Computer-Aided Verification

∗Modul „Logic and Computability“ (6,0 ECTS)

6,0/4,0 VU Logic and Computability

∗Modul „Program Analysis“ (6,0 ECTS)

6,0/4,0 VU Program Analysis

∗Modul „SAT Algorithms, Applications and Extensions“ (6,0 ECTS)

6,0/4,0 VU SAT Algorithms, Applications and Extensions

∗Modul „Advanced Topics In Verification and Automated Reasoning“ (min. 3,0 ECTS)

Prüfungsfach „Methods in Computer Science“

Modul „Seminar in Computer Science“ (min. 3,0 ECTS)

3,0/2,0 SE Seminar in Computer Science

∗Modul „Project in Computer Science“ (6,0–12,0 ECTS)

6,0/4,0 PR Project in Computer Science 1
6,0/4,0 PR Project in Computer Science 2

Prüfungsfach „Extension“

*Modul „Extension“ (up to 12,0 ECTS)

Prüfungsfach „Freie Wahlfächer und Transferable Skills“

Modul „Freie Wahlfächer und Transferable Skills“ (9,0 ECTS)

Prüfungsfach „Diplomarbeit“

1,5/1,0 SE	Seminar für Diplomand_innen
27,0 ECTS	Diplomarbeit
1,5 ECTS	Kommissionelle Abschlussprüfung

E Wahlfachkatalog „Transferable Skills“

Die Lehrveranstaltungen, die im Modul *Freie Wahlfächer und Transferable Skills* aus dem Themenbereich „Transferable Skills“ zu wählen sind, können unter anderem aus dem folgenden Katalog gewählt werden.

3,0/2,0 SE Coaching als Führungsinstrument 1
3,0/2,0 SE Coaching als Führungsinstrument 2
3,0/2,0 SE Didaktik in der Informatik
1,5/1,0 VO EDV-Vertragsrecht
3,0/2,0 VO Einführung in die Wissenschaftstheorie I
3,0/2,0 VO Einführung in Technik und Gesellschaft
3,0/2,0 SE Folgenabschätzung von Informationstechnologien
3,0/2,0 VU Forschungsmethoden
3,0/2,0 VO Frauen in Naturwissenschaft und Technik
3,0/2,0 SE Gruppendynamik
3,0/2,0 VU Kommunikation und Moderation
3,0/2,0 SE Kommunikation und Rhetorik
1,5/1,0 SE Kommunikationstechnik
3,0/2,0 VU Kooperatives Arbeiten
3,0/2,0 VU Präsentation und Moderation
1,5/1,0 VO Präsentation, Moderation und Mediation
3,0/2,0 UE Präsentation, Moderation und Mediation
3,0/2,0 VU Präsentations- und Verhandlungstechnik
4,0/4,0 SE Privatissimum aus Fachdidaktik Informatik
3,0/2,0 VU Rhetorik, Körpersprache, Argumentationstraining
3,0/2,0 VU Softskills für TechnikerInnen
3,0/2,0 VU Techniksoziologie und Technikpsychologie
3,0/2,0 VO Theorie und Praxis der Gruppenarbeit
3,0/2,0 VO Zwischen Karriere und Barriere

F Erweiterungsstudium Innovation

Studierende, die ihre im Masterstudium erworbenen Kompetenzen für die Gründung eines Startups bzw. im Management eines Unternehmens oder für Projekttätigkeit im universitären Umfeld anwenden wollen, können die für diese Tätigkeiten notwendigen zusätzlichen Kompetenzen im Rahmen des Erweiterungsstudiums *Innovation* erwerben, welches begleitend zum Masterstudium absolviert werden kann.

Der (zusätzliche) Arbeitsaufwand für das englischsprachige Erweiterungsstudium *Innovation* beträgt 30 ECTS-Punkte (dies entspricht einem Semester). Der Abschluss des Erweiterungsstudiums *Innovation* kann auch noch nach Abschluss des Masterstudiums erfolgen.