

CORSO DI LAUREA IN INFORMATICA **BACHELOR IN INFORMATIK**

Contenuto degli insegnamenti **Inhalt der Lehrveranstaltungen**

Analysis <ul style="list-style-type: none"> Sequences and series Univariate functions Derivatives, differentials and Taylor Theorem Riemann integral Logarithmic and exponential functions Limits of functions and continuity
Artificial Intelligence
Module 1: Foundation of Artificial Intelligence <ul style="list-style-type: none"> Artificial Intelligence and Agents Searching for Solutions Reasoning with Constraints Propositions and inference Planning with Certainty Multiagent Systems and Games
Module 2: Machine Learning in Practice <ul style="list-style-type: none"> Feature Extraction Frequent Pattern Recognition Regression Analysis Rule-based Classification and Decision Trees Bayesian Classifiers K-Means Clustering
Computational Security
Module 1: Computational Mathematics <ul style="list-style-type: none"> Principles of finite precision computation Direct methods for solving linear systems Iterative methods for linear algebra Singular value decomposition Rootfinding methods for solving nonlinear equations

<ul style="list-style-type: none"> • Functional approximation
<p>Module 2: Information Security</p> <ul style="list-style-type: none"> • Basic definitions: CIA, threat, attack, vulnerability, access control • Risk assessment • Basics of cryptography • Network attack and defense • Usability • Security policies
<p>Computer Networks</p> <ul style="list-style-type: none"> • Introduction to computer networks • ISO OSI reference model • Internet applications and application protocols (HTTP, SMTP, DNS) • Network protocols: TCP/IP, Ethernet • Sockets and RPCs • Failure robustness, security
<p>Computer Programming</p> <ul style="list-style-type: none"> • Basic algorithms and data structures • Data types and expressions • Classes and objects • Conditionals and loops • Object-oriented design • Arrays and collections • Input/Output and exception handling • Inheritance and polymorphism • Recursion
<p>Computer Systems Architecture</p> <ul style="list-style-type: none"> • Computer systems organization: processors, primary memory, secondary memory, input/output and parallel architectures. • Boolean algebra and gates: Boolean algebra, gates, implementation of Boolean functions, circuit equivalence. • Digital circuits: arithmetic circuits, clocks, memory, CPU chips, buses. • Microarchitecture: design of the microarchitecture level, performance optimization. • Instruction sets: data types, instruction formats, addressing, instruction types, flow of control. • Introduction to Assembly language
<p>Database Management Systems</p> <ul style="list-style-type: none"> • Physical data storage • Indexing and hashing • Query processing and optimization • Transaction processing • Concurrency control • Recovery
<p>English for Computer Scientists 1</p>

<ul style="list-style-type: none"> • General overview of grammatical structures at the C1 level; • Development of receptive skills through the exposure to and analysis of various types of written and spoken discourse typical in Computer Science and development of grammatical and lexical range and accuracy so that communication is fluent and spontaneous; • Vocabulary acquisition and word-building techniques; lexicogrammar.
<p>English for Computer Scientists 2</p> <ul style="list-style-type: none"> • Study skills: focus on developing the specific study skills that are required in this undergraduate programme in Computer Science including critical thinking skills; • Writing skills: practice of coherent academic discourse to produce subject-specific texts in English at the C1 level, including formal academic emails, reports and summaries; • Spoken skills: improvement of spoken interaction and production through the practice and production of academically and professionally acceptable presentations and other domain-specific speaking activities.
<p>Formal Languages and Compilers</p> <ul style="list-style-type: none"> • Formal language theory • Regular languages: automata, regular expressions, regular grammars • Context free languages (stack machines) • Lexical and syntactic analysis: Lexer specification, top-down and bottom-up parsing • Semantic analysis: Rules for type checking, symbol table and control flow • Intermediate code generation
<p>German for Computer Scientists</p> <ul style="list-style-type: none"> • Listening skills: comprehension of talks, documentary, reportings, descriptions in different contexts, on different media, about ICT topics • Writing skills: practice of coherent academic discourse to produce subject-specific texts (for example application letter, report, product review, compliant mail, instructions, essay, abstract, summary, seminar work etc.) about ICT topics; • Spoken skills: improvement of spoken interaction and production through the practice and production of academically and professionally acceptable presentations and other domain-specific speaking activities; • Development of receptive skills (reading and listening, both global and detailed) through the exposure to and analysis of various types of authentic written and spoken discourse typical in Computer Science and development of grammatical and lexical range and accuracy so that communication is fluent and spontaneous; • Language mediation (mediating communication, text and concepts) from English to German and viceversa about area of expertise (ICT); • Vocabulary acquisition and word-building techniques; lexicogrammar.
<p>Intelligent Agents</p>
<p>Module 1: Knowledge Representation</p> <ul style="list-style-type: none"> • Individuals and Relations • Knowledge Representation and Logic • Model Theory • Theorem Proving • Ontologies, DL, and Knowledge-Based Systems • Formal Reasoning and Common-Sense Knowledge

<p>Module 2: Intelligent Agents Project</p> <ul style="list-style-type: none"> • Overview of the main AI techniques: exact and approximate methods, handling imperfect information, use and model domain knowledge. • Tools for development of AI systems • Programming languages for AI • AI programming techniques • Projects on AI topics, such as: knowledge representations, games, automated planning, applications of constraint solving, multiagent systems
<p>Interactive Interface Design and Development (cohorts from 2023)</p> <ul style="list-style-type: none"> • Fundamentals of interaction design. • Fundamental interaction design principles for web apps. • Fundamental interaction design patterns for web apps. • Interaction design prototyping techniques for web apps. • Interaction design prototyping tools for web apps. • Fundamentals of web programming, client-side, for web apps
<p>Introduction to Business Administration (cohorts from 2020 to 2022)</p> <ul style="list-style-type: none"> • Introduction to management: company overview and business functions, capital configuration, company equilibrium, organizational dynamics • Accrual Accounting and Financial Statements • Managerial accounting for decision making (costing and pricing), planning, budgeting and reporting
<p>Italian for Computer Scientists</p> <ul style="list-style-type: none"> • Listening skills: comprehension of talks in different contexts, live, by phone or other media, about ICT topics • Writing skills: practice of coherent academic discourse to produce subject-specific texts; practice of all communication texts, such as e-mails, web texts; • Spoken skills: improvement of spoken interaction and production through the practice and production of academically and professionally acceptable presentations and other domain-specific speaking activities; • Development of receptive skills through the exposure to and analysis of various types of written and spoken discourse typical in ICT and development of grammatical and lexical range and accuracy so that communication is fluent and spontaneous; • Language mediation (mediating communication, text and concepts) from English to German and viceversa about area of expertise (ICT); • Vocabulary acquisition and word-building techniques; lexicogrammar.
<p>Linear Algebra</p> <ul style="list-style-type: none"> • Background on complex numbers, trigonometry and polynomials • Vectors and matrices: • Linear Systems • vector spaces: • Linear operators • Spectral analysis
<p>Maker Lab (cohorts from 2020 to 2022)</p> <ul style="list-style-type: none"> • Basics of programming for physical computing and interactions with the world using Python 3 • Basics of electronics for physical computing: interruptors, sensors, actuators and hats

<ul style="list-style-type: none"> Basics of interaction design and development for physical computing
Mobile and Physical Systems
<p>Module 1: Engineering of Mobile Systems</p> <ul style="list-style-type: none"> Functional and declarative programming Design of mobile applications Frameworks and platforms for mobile development Data and resource management in a mobile context Mobile device sensors Internet of Things
<p>Module 2: Physical Computing Project (cohorts from 2020 to 2022) / Prototyping Physical Interactive Experiences (cohorts from 2023)</p> <ul style="list-style-type: none"> Introduction to interaction design for physical computing Physical computing hardware for interactive solutions Physical computing software for interactive solutions Ideation and conceptualisation of physical computing solutions Development of physical computing solutions Evaluation of physical computing solutions
<p>Operating Systems</p> <ul style="list-style-type: none"> Programming in C Scheduling and concurrency Processes and synchronization File systems and memory management Storage management Security and protection
<p>Probability Theory and Statistics</p> <ul style="list-style-type: none"> Basic concepts: probability spaces, conditional probability, Bayes' Theorem, independent events Random variables: distribution, density, expectation, variance, covariance, law of large numbers Special distributions: Bernoulli, Binomial, Poisson, Exponential, Normal, Chi-Square, t-Distribution Sampling: sums of random variables, central limit theorem, sample variance Parameter Estimation: maximum likelihood estimates, interval estimates, confidence intervals Hypothesis testing: significance levels, test statistics, p-values
<p>Project and Teamwork Management</p> <ul style="list-style-type: none"> Project and team work management methods and techniques: goal specification techniques, coordination and collaboration techniques, performance and risk management Human resources management: communication, conflict management Tool support for project and team work management
<p>Scientific Writing and Communication</p> <ul style="list-style-type: none"> Planning communication: audiences, media and presenters In-reach communication: structure of thesis, journal paper, posters, presentation Out-reach communication: technical manuals, writing for the web; press release; social media; public dissemination

Software Engineering

- Software life-cycle: principles and methodologies
- Software processes and software project management
- Requirements engineering: elicitation and modeling
- System modeling and construction: UML, design patterns
- Software testing: principles and techniques
- Software management and evolution

Software Architecture (cohort from 2020 to 2022) Software Systems Engineering (cohort from 2023)

Module 1: Software Systems Architecture

- Software and systems architecture principles
- Architecture process and activities: specification, validation
- Architectural description and modeling
- Stakeholders and viewpoints
- Quality considerations: security, performance, modifiability
- Patterns of systems architectures

Module 2: Tools and Techniques for Software Testing

- Techniques for black box and white box testing
- Automated testing
- Dynamic Testing
- Static testing
- Performance and monitoring
- Introduction to search-based testing

Web and Internet Engineering

- Development of web applications: basics of usability, accessibility and responsive design
- Web protocols and markup languages
- Client-side dynamicity and web scripting languages
- Client-side GUI frameworks
- Web application design and web services
- Languages and frameworks for server-side web development