

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

**Contenuto degli insegnamenti**  
**Inhalt der Lehrveranstaltungen**

<b>Primo anno / Erstes Jahr</b>
<p>Linear Algebra</p> <ul style="list-style-type: none"> <li>• Background on complex numbers, trigonometry and polynomials</li> <li>• Vectors and matrices:</li> <li>• Linear Systems</li> <li>• vector spaces:</li> <li>• Linear operators</li> <li>• Spectral analysis</li> </ul>
<p>Analysis</p> <ul style="list-style-type: none"> <li>• Sequences and series</li> <li>• Univariate functions</li> <li>• Derivatives, differentials and Taylor Theorem</li> <li>• Riemann integral</li> <li>• Logarithmic and exponential functions</li> <li>• Normed vector spaces</li> </ul>
<p>Discrete Mathematics</p> <ul style="list-style-type: none"> <li>• Elements of logic, propositions and quantifiers, methods of mathematical proof</li> <li>• Numbers and basic number theory</li> <li>• Set Theory, Russell Paradox and Halting Problem</li> <li>• Functions, infinite cardinalities and countability</li> <li>• Relations, orders, equivalence classes</li> <li>• Graphs and trees</li> </ul>
<p>Computer Programming</p> <ul style="list-style-type: none"> <li>• Basic algorithms and data structures</li> <li>• Data types and expressions</li> <li>• Classes and objects</li> <li>• Conditionals and loops</li> <li>• Object-oriented design</li> <li>• Arrays and collections</li> </ul>

- Input/Output and exception handling
- Inheritance and polymorphism
- Recursion

#### Computer Systems Architecture

- 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

#### Operating Systems

- Programming in C
- Scheduling and concurrency
- Processes and synchronization
- File systems and memory management
- Storage management
- Security and protection

#### English for Computer Scientists 1

- 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.

#### English for Computer Scientists 2

- 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.

### Secondo anno / Zweites Jahr

#### Maker Lab

- 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
- Basics of interaction design and development for physical computing

#### Database Management Systems

- Physical data storage
- Indexing and hashing
- Query processing and optimization

- Transaction processing
- Concurrency control
- Recovery

#### Formal Languages and Compilers

- 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

#### Introduction to Databases

- Relational data model and relational algebra
- The SQL language
- Using SQL in database applications: API, embedded SQL
- The Entity Relationship model
- Conceptual database design
- Logical database design

#### Programming Project

- Memory models in Java
- Virtual functions, late binding, overriding, and overloading
- Exception handling
- Reflection and runtime type identification
- Generics and templates
- I/O, serialization and XML/JSON processing
- Designing large applications: design patterns, advanced GUI
- Multithreading
- Code optimization

#### 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

#### 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

## Computer Networks

- 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

## Probability Theory and Statistics

- 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

## Software Architecture

### 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

## Artificial Intelligence

### Module 1: Foundation of Artificial Intelligence

- 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

- Feature Extraction
- Frequent Pattern Recognition

- Regression Analysis
- Rule-based Classification and Decision Trees
- Bayesian Classifiers
- K-Means Clustering

#### German for Computer Scientists

- 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.

#### Italian for Computer Scientists

- 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.

### **Terzo anno / Drittes Jahr**

#### Computational Security

##### Module 1: Computational Mathematics

- 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
- Functional approximation

##### Module 2: Information Security

<ul style="list-style-type: none"> <li>• Basic definitions: CIA, threat, attack, vulnerability, access control</li> <li>• Risk assessment</li> <li>• Basics of cryptography</li> <li>• Network attack and defense</li> <li>• Usability</li> <li>• Security policies</li> </ul>
<p>Project and Teamwork Management</p> <ul style="list-style-type: none"> <li>• Project and team work management methods and techniques: goal specification techniques, coordination and collaboration techniques, performance and risk management</li> <li>• Human resources management: communication, conflict management</li> <li>• Tool support for project and team work management</li> </ul>
<p>Introduction to Business Administration</p> <ul style="list-style-type: none"> <li>• Introduction to management: company overview and business functions, capital configuration, company equilibrium, organizational dynamics</li> <li>• Accrual Accounting and Financial Statements</li> <li>• Managerial accounting for decision making (costing and pricing), planning, budgeting and reporting</li> </ul>
<p>Scientific Writing and Communication</p> <ul style="list-style-type: none"> <li>• Presentation techniques: structure of presentations, interacting with PowerPoint, slide design, body language and positioning, presentation of participants, feedback</li> <li>• Communication techniques: structure of presentations, interacting with PowerPoint, slide design, body language and positioning, presentation of participants, feedback</li> <li>• Scientific writing: academic language, structure of scientific documents, scientific sources, thesis writing</li> </ul>
<p>Mobile and Physical Systems</p>
<p>Module 1: Engineering of Mobile Systems</p> <ul style="list-style-type: none"> <li>• Functional and declarative programming</li> <li>• Design of mobile applications</li> <li>• Frameworks and platforms for mobile development</li> <li>• Data and resource management in a mobile context</li> <li>• Mobile device sensors</li> <li>• Internet of Things</li> </ul>
<p>Module 2: Physical Computing Project</p> <ul style="list-style-type: none"> <li>• Introduction to interaction design for physical computing</li> <li>• Physical computing hardware for interactive solutions</li> <li>• Physical computing software for interactive solutions</li> <li>• Ideation and conceptualisation of physical computing solutions</li> <li>• Development of physical computing solutions</li> <li>• Evaluation of physical computing solutions</li> </ul>
<p>Intelligent Agents</p>
<p>Module 1: Knowledge Representation</p> <ul style="list-style-type: none"> <li>• Individuals and Relations</li> </ul>

- Knowledge Representation and Logic
- Model Theory
- Theorem Proving
- Ontologies and Knowledge-Based Systems
- Planning with Individuals and Relations

#### Module 2: Intelligent Agents Project

- Overview of the main AI techniques: exact and approximate methods, handling imperfect information, use and model domain knowledge.
- Tools for development of AI systems
- Functional and Logic Programming languages for AI
- AI programming techniques
- Projects on AI topics, such as: knowledge representations, games, automated planning, applications of constraint solving, multiagent systems