



**Offer of classes in 2026/2027  
Computer science, Warsaw**

**1 semester**

Linear Algebra	6 ECTS
Elements of Logic and Set Theory	6 ECTS
Introduction to Programming	6 ECTS
Developer Tools	3 ECTS
Social and Group Processes, Communication in Organizations	4 ECTS

**2 semester**

Theoretical Foundations of Computer Science	5 ECTS
Discrete Mathematics	6 ECTS
Mathematical Analysis 1	6 ECTS
Programming in C	5 ECTS
Computer Architecture and Organization	5 ECTS

**3 semester**

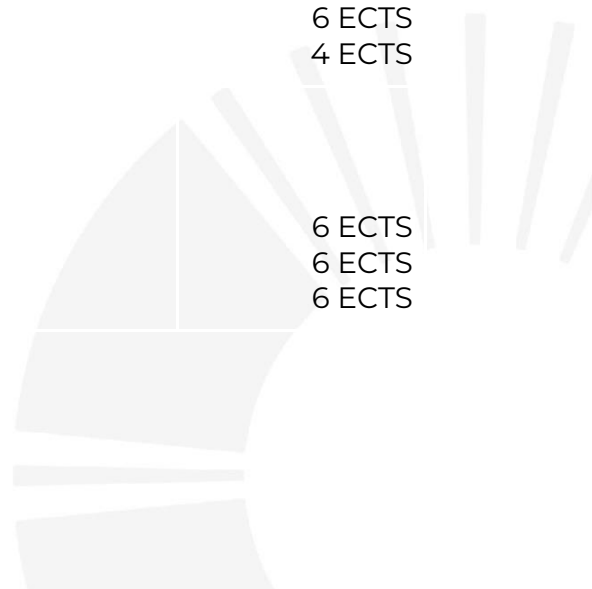
Object-Oriented Programming	5 ECTS
Operating Systems	5 ECTS
Mathematical Analysis 2	5 ECTS
Probability and Statistics	5 ECTS
Basics of Entrepreneurship and Intellectual Property Protection	3 ECTS

**4 semester**

Algorithms and Data Structures	6 ECTS
Data Analysis Techniques	5 ECTS
Computer Networks	6 ECTS
Elements of Cryptography and Number Theory	6 ECTS
Representations, Cognitive Processes and Learning	4 ECTS

**5 semester**

Databases	6 ECTS
Programming Languages	6 ECTS
Numerical Methods	6 ECTS





**Uniwersytet  
SWPS**

Software Engineering	5 ECTS
Soft Skills in the Work of an IT Specialist	3 ECTS

**6 semester**

Machine Learning in Data Analysis	4 ECTS
-----------------------------------	--------





**Offer of classes in 2026/2027  
Computer science, Warsaw**

**1 semester**

**Linear Algebra**

6 ECTS

**Elements of Logic and Set Theory**

6 ECTS

The aim of the course is to introduce students to the fundamental concepts of classical propositional and predicate logic as well as basic elements of set theory. The course provides students with theoretical foundations necessary for formal reasoning and for understanding selected mathematical structures and methods used in analytical thinking. Students become familiar with the principles of classical propositional and predicate logic and learn how logical expressions are constructed and interpreted. The course also introduces key concepts of set theory, including basic operations on sets, the Cartesian product, and the power set. In addition, students explore selected elements of the theory of binary relations, the concept of functions understood as particular types of relations, and the notion of cardinality of sets. The course also presents the methods of mathematical induction and recursion as fundamental tools used for defining and analyzing mathematical objects. The aim of the course is to develop students' ability to use formal reasoning and apply logical and set-theoretical concepts in problem solving.

**Introduction to Programming**

6 ECTS

**Developer Tools**

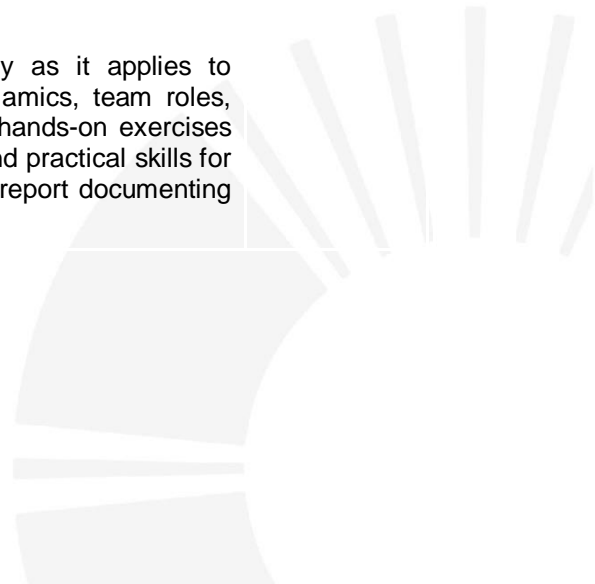
3 ECTS

The subject is designed to familiarise students with the basic tools of the job. The course aims to provide basic knowledge of the programmer's work environment, especially in Linux systems. Students will understand the issues and benefits related to application versioning and virtualization, using containerization as an example. Furthermore, the knowledge gained allows students to search for sources for further development. During the course, the student will become familiar with Linux and basic Bash shell commands, allowing them to navigate freely in this type of system, install packages, read and edit files, etc. Through laboratory exercises, participants will learn to use basic shell commands and write and run a simple Bash script. Further on, students will understand the basics of a version control system using Git as an example, enabling them to create and manage a project repository successfully. Additionally, the curriculum covers how to create a configuration file to create and run a container with a low level of complexity. Ultimately, the knowledge and skills acquired enable a critical approach to the content received and effective verification of sources.

**Social and Group Processes, Communication in Organizations**

4 ECTS

This course introduces Computer Science students to social psychology as it applies to organizational and team settings. It covers key topics such as group dynamics, team roles, interpersonal communication, and project management, complemented by hands-on exercises using collaboration tools. Students develop both theoretical understanding and practical skills for working effectively in teams. The course is assessed through an individual report documenting participation in group projects.





**2 semester**

**Theoretical Foundations of Computer Science**

5 ECTS

**Discrete Mathematics**

6 ECTS

**Mathematical Analysis 1**

6 ECTS

The aim of the *Mathematical Analysis* course is to introduce students to the fundamental concepts and methods of classical calculus, with particular emphasis on their practical applications in computer science and related fields. The course provides a structured and comprehensive presentation of the foundations of analysis, enabling students to understand and apply the core notions of limits, derivatives, and integrals of functions of one real variable. Throughout the course, students explore the nature and behavior of numerical sequences and real-valued functions, focusing on their limits, continuity, and differentiability. Considerable attention is devoted to the basic differentiation rules, the geometric interpretation of the derivative, and the role of derivatives in solving analytical problems—such as identifying local extrema, constructing linear approximations, and analyzing the shapes of graphs. An essential component of the course is the study of higher-order derivatives and the Taylor theorem. Students learn how Taylor expansions can be applied to approximation methods and numerical integration, gaining insight into one of the central tools of computational mathematics. The course also encompasses a rigorous introduction to indefinite integrals and the definite Riemann integral, including classical techniques of integration and the evaluation of integrals through numerical approaches. Moreover, the course highlights key applications of the Riemann integral, including computing areas between curves, the length of a curve, and the volume and surface area of solids generated by rotation. These applications demonstrate the relevance of analytical methods to real-world mathematical modeling. In addition to acquiring technical skills, students develop the ability to critically assess their own understanding of problem statements, recognize gaps in reasoning, and select appropriate analytical tools to solve problems that commonly arise in computer science.

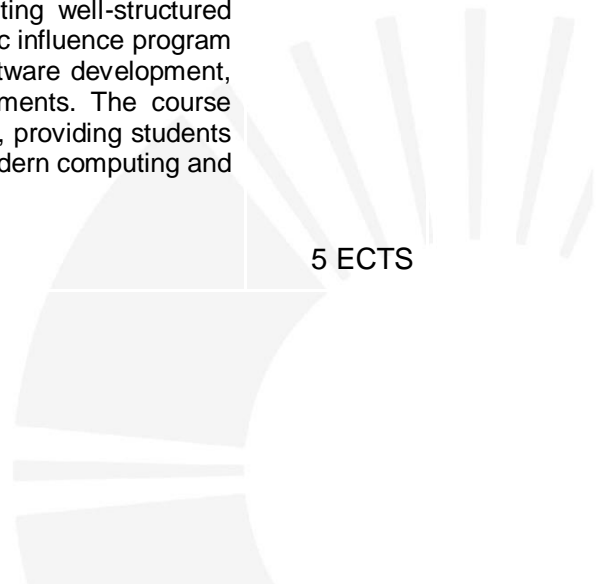
5 ECTS

**Programming in C**

The aim of this course is to provide a systematic and advanced understanding of the C programming language, bridging the gap between fundamental computer science concepts and practical software engineering. Students will begin by mastering the core elements of the language, including data types, operators, and control structures. The course then progresses to more advanced topics such as pointers, pointer arithmetic, and memory management, enabling students to develop a strong foundation in low-level programming. Through the study of modular program design, C strings, and structured data types, students will gain practical skills for developing efficient and scalable applications. Emphasis is placed on writing well-structured programs and understanding how memory organization and architectural logic influence program performance in C. In addition, the course highlights practical aspects of software development, including the configuration and use of professional programming environments. The course concludes with advanced program structures and file input/output operations, providing students with a comprehensive understanding of C and its continued importance in modern computing and systems programming.

5 ECTS

**Computer Architecture and Organization**





The course aims to provide students with comprehensive knowledge of computer system architectures, their structural components, and operational principles, including both their capabilities and constraints. The lectures introduce students to the physical foundations of information processing, binary logic, computer arithmetic, and the design principles of various computer architectures. Particular emphasis is placed on the fundamental concepts of virtualization and cloud computing technologies. Within the course, students explore topics such as memory hierarchies, computer I/O devices, microcontrollers (using Arduino as an example), and modern processor architectures, including CISC (x86) and RISC (RISC-V) approaches, as well as multiprocessor and multi-core systems. The laboratory classes involve solving practical tasks, including writing simple programs that cover topics discussed in the lectures, and managing the computer system environment. Students develop proficiency in selecting and justifying appropriate low-level programming tools—such as the basics of RISC-V assembly programming—to solve specific IT problems. Furthermore, participants acquire practical skills in GNU/Linux system administration and competently apply virtualization and cloud computing technologies. The course also develops students' ability to critically evaluate information, verify the credibility of sources, and demonstrate professional responsibility and a commitment to ethical principles in computer science practice.

### **3 semester**

#### **Object-Oriented Programming**

5 ECTS

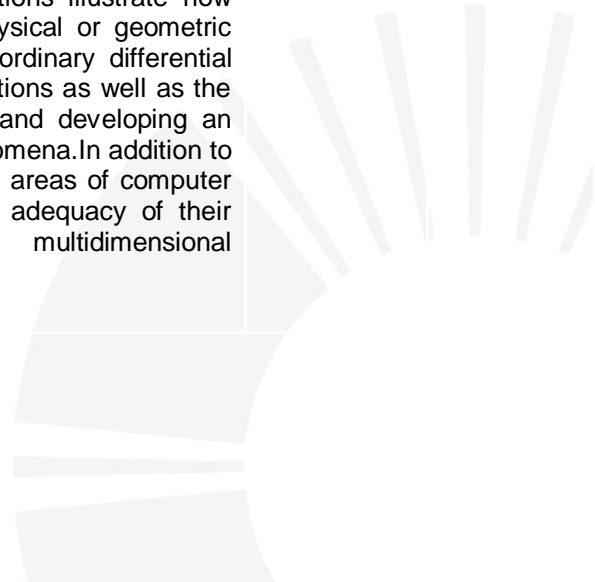
#### **Operating Systems**

5 ECTS

#### **Mathematical Analysis 2**

5 ECTS

The aim of the *Mathematical Analysis II* course is to introduce students to the key concepts and techniques of multivariable calculus and introductory differential equations. Building on the foundations developed in *Mathematical Analysis I*, the course extends students' understanding of limits, derivatives, and integrals to functions of several real variables and explores the analytical tools required to solve multidimensional and dynamic problems encountered in mathematics, computer science, and engineering. Throughout the course, students examine the behavior of multivariable functions by analyzing their limits, continuity, and differentiability. Special emphasis is placed on partial and directional derivatives, the gradient vector, and the geometric interpretation of differentiability through tangent planes. Students develop the ability to identify and classify local extrema of multivariable functions using standard analytical techniques, including second-order tests. A central part of the course concerns double and triple integrals over regions in the plane and in space. Students learn how to evaluate these integrals over normal and regular domains, and how coordinate transformations—specifically polar, cylindrical, and spherical coordinates—simplify computations involving symmetry or geometric structure. Applications illustrate how multivariable integration is used to compute mass, volume, and other physical or geometric quantities. The course also provides a rigorous introduction to first-order ordinary differential equations. Students study linear homogeneous and nonhomogeneous equations as well as the Bernoulli equation, gaining experience with classical solution techniques and developing an appreciation for the role of differential equations in modeling real-world phenomena. In addition to mastering technical skills, students learn to translate problems from various areas of computer science into precise mathematical language, and to critically assess the adequacy of their analytical approach when solving complex multidimensional tasks.





### **Probability and Statistics**

5 ECTS

This course covers the most important theoretical concepts in probability theory and mathematical statistics. Its goal is to introduce students to probability theory for both countable and uncountable probability spaces, including: elements of combinatorics, the concept of probability in discrete spaces, properties of probability, computing probability based on geometric intuitions for uncountable spaces, the theory of random variables (including two-dimensional ones), limit theorems, most important probability distributions, and key topics in descriptive and mathematical statistics: determining the central tendency of a series of numerical data, point and interval estimation, and the theory of statistical hypothesis testing.

### **Basics of Entrepreneurship and Intellectual Property Protection**

3 ECTS

This course covers the fundamentals of entrepreneurship and intellectual property for Computer Science students. Topics range from the mindset and responsibilities of an entrepreneur, through practical tools like the Business Model Canvas, to understanding markets, competition, and innovation typologies. The course also addresses intellectual property law, focusing on how it applies in the technology industry and its practical limitations. It is assessed through a single end-of-semester test based on lectures and required readings.

### **4 semester**

### **Algorithms and Data Structures**

6 ECTS

The Algorithms and Data Structures course provides students with a solid foundation in the mathematical principles and analytical techniques required for evaluating algorithmic efficiency. Students are introduced to fundamental asymptotic notations ( $O$ ,  $\Omega$ , and  $\Theta$ ) which are used to analyze and compare algorithms in terms of time and space complexity. The course covers essential algorithm design paradigms, including divide-and-conquer, dynamic programming, and greedy algorithms, and applies these approaches to fundamental computational problems such as sorting, searching, and graph processing. In addition, students study the structure and operation of core data structures, including stacks, queues, linked lists, binary search trees, heaps, and graphs. The course focuses on understanding how these structures support efficient data organization and algorithm implementation. Practical components of the course require students to select appropriate algorithms for specific problems and implement their solutions using the C programming language. Topics such as Breadth-First Search (BFS), Depth-First Search (DFS), and shortest path algorithms are explored to develop students' ability to design efficient solutions for complex computational tasks

### **Data Analysis Techniques**

5 ECTS

This course introduces the fundamentals of statistical data processing, data analysis, visualization, and modelling. It is designed to help students understand how data can be collected, prepared, explored, and transformed in order to support further analytical tasks and machine learning applications. The course combines theoretical foundations with practical elements, giving students both conceptual understanding and hands-on familiarity with commonly used tools for data analysis. The course covers essential topics related to data preprocessing, statistical description of data, exploratory data analysis, and basic modelling techniques. Students will learn how to work with datasets, identify patterns and dependencies, visualize data in a clear and informative way, and interpret analytical results. They will also become familiar with widely used libraries and tools that support data processing and analysis in practical work. An important part of the course is preparing data for machine learning. Students will learn how to clean and organize data, handle missing values, transform variables, and prepare datasets so that they can be effectively used for training machine learning models. By the end of the course, students should understand the role



of proper data preparation in the overall data analysis pipeline and in the development of reliable intelligent systems.

### **Computer Networks**

6 ECTS

### **Elements of Cryptography and Number Theory**

6 ECTS

The aim of the course is to introduce students to selected topics in number theory as well as the basic concepts of modern cryptography. The course provides theoretical foundations necessary for understanding the mathematical mechanisms used in cryptographic systems and methods of securing information in computer systems. Students become familiar with the basic concepts of number theory, in particular the properties of prime numbers, the extended Euclidean algorithm, and Euler's and Fermat's theorems together with their applications. The course also covers congruences and systems of congruences, the Chinese Remainder Theorem, and selected applications of modular arithmetic. In the later part of the course, students learn the fundamental principles of selected symmetric and asymmetric encryption methods as well as the mathematical foundations underlying their construction. Particular attention is given to the role of number theory in the design and analysis of cryptographic systems. The course aims to develop students' analytical thinking skills and their understanding of the mathematical foundations of modern methods of information protection.

### **Representations, Cognitive Processes and Learning**

4 ECTS

This course offers a comprehensive introduction to cognitive psychology, examining how the human mind perceives, attends to, remembers, and processes information. Lectures explore core topics including perception, attention, memory, concepts and representations, imagination, reasoning, and learning. Exercises reinforce these themes through applied work on topics like problem-solving and decision-making. Together, the course builds both theoretical knowledge and practical insight into the cognitive processes underlying human behaviour and learning.

## **5 semester**

### **Databases**

6 ECTS

### **Programming Languages**

6 ECTS

### **Numerical Methods**

6 ECTS

The course aims to provide students with a solid understanding of fundamental numerical analysis concepts, alongside the methods, techniques, and tools used to solve essential numerical problems. The lectures introduce students to the core issues of numerical tasks, including the properties of floating-point arithmetic, the numerical stability of algorithms, and vector and matrix norms. Within the course, students explore both direct and iterative methods for solving systems of linear equations (such as Gaussian elimination, Cholesky, Jacobi, Gauss-Seidel, and SOR methods), as well as techniques for calculating determinants, inverse matrices, and condition numbers. Further topics cover finding eigenvalues and eigenvectors, solving nonlinear equations (e.g., bisection, secant, and Newton's methods), single-variable function interpolation (Lagrange, Newton, and Hermite forms), and numerical integration (including Newton-Cotes and Gauss quadratures). During practical classes and computer laboratories, students apply their theoretical knowledge by solving tasks related to the current lecture topics and by implementing selected



methods and algorithms. Participants develop the ability to evaluate the suitability of classic and contemporary tools, plan and conduct computer simulations, and critically assess the quality of the obtained results. Furthermore, the classes focus on equipping students with the skills to select the appropriate method for a specific problem, verify necessary assumptions, and critically evaluate their own reasoning and understanding of complex numerical challenges.

**Software Engineering**

5 ECTS

**Soft Skills in the Work of an IT Specialist**

3 ECTS

**6 semester**

**Machine Learning in Data Analysis**

4 ECTS

This course introduces students to current trends and selected practical applications of machine learning. It presents the main types of machine learning problems and explains how they can be addressed using supervised learning, unsupervised learning, and reinforcement learning approaches. During the course, students will learn how to identify tasks that can be solved with machine learning methods, choose appropriate techniques, and understand their strengths, limitations, and typical areas of application. The course covers fundamental and widely used machine learning techniques, including neural networks, decision trees, and regression-based methods. Students will also become familiar with basic concepts related to data preparation, model training, evaluation, and interpretation of results. Particular attention is given to understanding how machine learning models work in practice and how they can be applied to solve real-world problems in different domains. The course combines theoretical foundations with practical examples, allowing students to build an overall understanding of modern machine learning methods and their role in contemporary data analysis and intelligent systems.

