



**UNIwersYTET
WSB MERITO
GDAŃSK**

WSB Merito University in Gdańsk Faculty of
Computer Science and New Technologies

Study programme For
the field of study
"Computer Science"
First-cycle studies

Studies: full-time - hybrid Profile: practical

Academic year 2025/2026

I. GENERAL CHARACTERISTICS OF THE FIELD OF STUDY

| | | |
|--|---|-------------------------------|
| Name of the field of study | Computer Science | |
| Level of education (first-cycle studies / second-cycle studies / uniform master's studies) | First-cycle studies | |
| Education profile | Practical | |
| Form of studies full-time/part-time | Full-time Hybrid | |
| Duration of studies (in semesters) | 7 semesters | |
| Total number of ECTS credits for a given form of study. | 210 | |
| Total number of hours specified in the programme of study | Full-time studies 5320 hours | Part-time studies - |
| Professional title awarded to graduates | Engineer | |
| Duration of professional practice | 960 hours | |
| Language of instruction | English | |
| Year of commencement of the educational cycle | 2025/2026 | |

II. LEARNING OUTCOMES

| Learning outcome symbol | Description of learning outcomes | Universal characteristic code |
|---|--|--------------------------------------|
| Knowledge The graduate knows and understands | | |
| INF_I_inž_W01 | advanced issues in mathematics, including the fundamentals of mathematical analysis, algebra, discrete mathematics (elements of logic and set theory, combinatorics and graph theory), probabilistic methods and statistics | P6S_WG |
| INF_I_Eng_W02_Eng | advanced issues in the field of programming (paradigms, languages, environments, methods, devices, etc.), taking into account the processes occurring in the life cycles of objects and IT devices | P6S_WG |
| INF_I_Eng_W03 | advanced knowledge of data structures, computational complexity and algorithms | P6S_WG |
| INF_I_Eng_W04 | advanced knowledge of issues related to information management and security, issues related to databases: types, environments, systems, servers, methods of design, configuration and management, including issues related to data warehouses, business intelligence, big data, data centres, data mining, cloud computing, etc. | P6S_WG |

| | | |
|---|---|----------------|
| INF_I_inż_W05 | advanced knowledge of issues related to network technologies, including basic communication protocols communication protocols, security and network application development | P6S_WG |
| INF_I_Eng_W06 | advanced knowledge of methods of knowledge representation and processing, knowledge bases, artificial intelligence methods and techniques, expert systems, etc. | P6S_WG |
| INF_I_Eng_W07_Eng | advanced issues related to processes occurring in the life cycle of information systems and the architecture of computer systems and equipment, as well as issues related to embedded systems and the Internet of Things | P6S_WG |
| INF_I_Eng_W08 | advanced issues related to the principles of operating systems, including processes, concurrency, task scheduling and memory management | P6S_WG |
| INF_I_Eng_W09 | advanced methodology and best practices used in IT project and service management | P6S_WG |
| INF_I_Eng_W10_Eng | advanced knowledge of issues related to software engineering, design of software development tools and environments, IT project life cycle, software specification, validation and verification, software maintenance | P6S_WG |
| INF_I_Eng_W11 | legal and social aspects of computer science, including issues related to professional and ethical responsibility, codes of , intellectual property | P6S_WK |
| INF_I_Eng_W12 | the relationship between computer science and other disciplines, especially in the field of social sciences | P6S_WK |
| INF_I_inż_W13 | advanced information and communication technologies (ICT) in the areas of specialisation studied and IT equipment appropriate for the areas of their applications | P6S_WG |
| INF_I_Eng_W14 | advanced foreign language terminology used in the field of ICT | P6S_WG |
| INF_I_inż_W15 | advanced knowledge of issues related to their chosen specialisation, particularly in the field of ongoing projects | P6S_WG |
| INF_I_inż_W16 | principles of creating and developing various forms of individual entrepreneurship | P6S_WK |
| INF_I_inż_W17 | issues in the field of ecology and environmental protection, corporate social responsibility necessary for understanding non-technical conditions of engineering activity | P6S_WK |
| Skills the graduate is able to | | |
| INF_I_inż_U01 | obtain information from literature, knowledge bases, the Internet and other reliable sources (including foreign language sources), integrate it, interpret it, draw conclusions and form opinions | P6S_UW, P6S_UK |
| INF_I_inż_U02 | work individually and in a team, including the ability to manage one's time, make commitments and meet deadlines | P6S_UO |
| INF_I_inż_U03 | communicate using various techniques in a professional environment and in other environments, including in a foreign language at B2 level and using IT tools | P6S_UK |
| INF_I_inż_U04 | read and understand technical documentation and articles in trade journals published in Polish and other foreign languages | P6S_UK, P6S_UW |

| | | |
|-------------------|---|-----------------------|
| INF_I_inż_U05 | use basic theoretical knowledge to obtain data for the practical analysis of processes and phenomena and draw conclusions on this basis | P6S_UW |
| INF_I_Eng_U06 | prepare an oral and multimedia presentation in Polish and a foreign language on ICT technologies, in particular for the needs of ongoing projects | P6S_UK, P6S_UW |
| INF_I_Eng_U07 | assess the level of one's own knowledge and skills for the purpose of further education | P6S_UU |
| INF_I_Eng_U08 | cooperate and work in a project group, taking on various roles | P6S_UO |
| INF_I_Eng_U09_Eng | apply the mathematical, analytical and experimental models and methods learned to formulate, analyse and solve engineering problems related to computer science, interpret the results obtained and draw conclusions | P6S_UW |
| INF_I_Eng_U10_Eng | use high-level programming languages and appropriate IT tools and be able to use them to develop useful computer programmes and software for devices, be able to further critically analyse the functioning of the developed solutions and evaluate these solutions | P6S_UW |
| INF_I_Eng_U11_Eng | use typical programming tools (programming environments, installation and configuration programmes, programming libraries, framework software) in practice programming and to create computer simulations | P6S_UW |
| INF_I_Eng_U12 | design, analyse in terms of correctness and computational complexity, and program algorithms; using basic algorithmic techniques and data structures; able to implement an algorithm in a computer programme or IT system | P6S_UW |
| INF_I_Eng_U13 | apply software engineering principles to the implementation of programming projects | P6S_UW |
| INF_I_Eng_U14 | apply the principles of software testing engineering to plan, design, conduct tests of the designed and reporting and implementing corrections | P6S_UW |
| INF_I_Eng_U15 | build a simple database system, create a simple, secure web application using databases, be able to formulate queries to the database | P6S_UW |
| INF_I_Eng_U16_Eng | design, independently implement, configure and administer a local computer network, including a wireless network, selecting technical materials appropriate to the specifications of the task requiring the use of engineering standards and engineering standards | P6S_UW |
| INF_I_Eng_U17_Eng | use artificial intelligence methods in programming practice and recognise aspects of their application systemic and non-technical, including ethical aspects | P6S_UW |
| INF_I_Eng_U18_Eng | design and implement a graphical user interface for an application, taking into account the principles of human-computer communication, and evaluate the usability of such an interface by planning and conducting experiments, including quality measurements. | P6S_UW |
| INF_I_Eng_U19_Eng | design simple devices, objects or implement processes in accordance with a given specification, using appropriately selected methods, techniques, tools and materials; design and analyse a simple IT system, assess its correctness and make corrections | P6S_UW |
| INF_I_Eng_U20 | implement an IT system and ensure the security storage and transmission of data and software | P6S_UW |

| | | |
|---|--|----------------|
| INF_I_Eng_U21_Eng | select and apply appropriate models, methods and good practices of project management for the implementation of IT projects, and when identifying and formulating specifications for engineering tasks and solving them, perform a preliminary economic assessment of the proposed solutions | P6S_UW |
| INF_I_Eng_U22_Eng | analyse IT systems in terms of compliance with environmental standards and be able to select appropriate technologies taking into account the environmental aspects of engineering activities | P6S_UW |
| Social competences The graduate is prepared to | | |
| INF_I_inž_K01 | formulate questions to deepen their understanding of a given topic or find missing elements reasoning | P6S_KK |
| INF_I_inž_K02 | understanding and appreciating the importance of intellectual honesty in one's own actions and those of others; being prepared to act ethically | P6S_KR |
| INF_I_inž_K03 | use normative systems and selected norms and rules (legal, professional, ethical) when solving tasks | P6S_KR |
| INF_I_inž_K04 | formulating opinions on basic IT issues, being aware of the importance and understanding non-technical aspects and effects of engineering activities | P6S_KK, P6S_KR |
| INF_I_inž_K05 | planning and preparing the implementation of an IT project, including consideration of sources of funding for its implementation | P6S_KO |

III. CLASSES OR CLASS GROUPS, REGARDLESS OF THE FORM OF DELIVERY, TOGETHER WITH THE ASSIGNED LEARNING OUTCOMES AND CURRICULUM CONTENT ENSURING THE ACHIEVEMENT OF THESE OUTCOMES

| CLASSES OR GROUPS OF CLASSES | REFERENCE TO DIRECT LEARNING OUTCOMES | CURRICULUM CONTENT |
|------------------------------|---|--|
| Physics | INF_I_inž_W01, INF_I_inž_W12, INF_I_inž_U01, INF_I_inž_K01, INF_I_inž_K04 | Units of measurement in the SI system Multiples and submultiples Measurement error – error discussion Operations on approximate numbers Description of basic measuring instruments and devices. Mechanics and molecular physics: Methods of measuring the density of liquids and solids Sliding forces and coefficient of friction on an inclined plane Determination of moment of inertia Young's modulus Voice and its parameters Surface tension and its parameters Viscosity of materials Heat: Air humidity Specific heat of solids Heat of vaporisation Heat of fusion of solids |

| | | |
|-------------|--|---|
| | | <p>Linear expansion of solids Volumetric expansion of liquids Temperature, methods of determining temperature Thermal conductivity</p> <p>Electricity:</p> <p>Electrical potential</p> <p>Direct and alternating current Current intensity</p> <p>Electrical resistance</p> <p>Basic laws governing the flow of electric current Electrical instruments: resistors, current sources, ammeter, voltmeter Optics:</p> <p>Basic concepts</p> <p>Focal length and collecting power of lenses Light as a wave, wavelength</p> <p>Microscope</p> <p>Refraction of light in liquids and solids</p> |
| Mathematics | INF_I_inž_W01, INF_I_inž_U09_in ž, INF_I_inž_K01 | <p>Linear function, definition, properties, graph. Linear equations and inequalities.</p> <p>Absolute value, equations and inequalities with absolute value. Quadratic function, definition, properties, graph. Quadratic equations and inequalities.</p> <p>Viet's formulas and their applications.</p> <p>Polynomials, Bezout's theorem, division of polynomials.</p> <p>Theorem on rational roots of polynomials with integer coefficients.</p> <p>Solving polynomial equations and inequalities. Rational equations and inequalities.</p> <p>Exponential function, definition, properties, graphs. Formulas related to the exponential function.</p> <p>Exponential equations and inequalities.</p> <p>Logarithmic function, definition, properties, graphs. Formulas related to logarithms.</p> <p>Logarithmic equations and inequalities.</p> <p>The concept of a sequence, arithmetic and geometric sequences and their properties. Formulas for the sum of an arithmetic and geometric sequence.</p> <p>Recursively defined sequences.</p> <p>Trigonometric functions, their properties and graphs. Trigonometric formulas.</p> <p>Trigonometric equations and inequalities.</p> |
| Calculus | INF_I_inž_W01, INF_I_inž_U09_in ž, INF_I_inž_K01 | <p>The concept of a function, domain, codomain, range. Overview of the most important elementary functions and their properties. Solving equations and inequalities related to elementary functions. The concept of a sequence limit, limit properties, improper limits (plus and minus infinity). Operations on limits, theorems on the existence of limits. Limits of basic sequences. Methods for calculating limits of sequences. The concept of the limit of a function and one-sided limits of functions. Properties of limits of functions. Operations on limits. Methods for calculating limits of functions.</p> <p>Applications of limits of functions to finding asymptotes of function graphs. The concept of continuity of a function, properties of continuous functions.</p> <p>The concept of an indefinite integral and its properties, integrals of elementary functions. Integration using basic formulas, integration by substitution, integration by parts. The concept of a definite integral, geometric interpretation of a definite integral. Methods for calculating definite integrals. Calculating the surface areas and volumes of solids of revolution.</p> <p>Integrals of discontinuous functions. Improper integrals.</p> |

| | | |
|---|---|--|
| | | <p>Functions of several variables. Partial derivatives. Differentiability of functions of several variables. Second-order derivatives, mixed derivatives.</p> <p>Schwarz's theorem. Local extrema of functions of two variables, conditions for the existence of extrema. Examples of finding local extrema for functions of two variables. Determining the maximum and minimum values of functions of two variables on a closed and bounded set</p> <p>closed and bounded set. Introduction to differential equations.</p> |
| Discrete mathematics | <p>INF_I_inž_W01, INF_I_inž_U09_in ž, INF_I_inž_K01</p> | <p>Introduction: countable sets</p> <p>Induction and recursion. Simple applications (recursive sequences, e.g. Fibonacci, merge-sort algorithms, GCD, Pythagorean tree) Number theory. Definitions of div, mod, congruence relation, applications using examples of checking digit calculation and number validation (bank accounts, ISBN, REGON, etc.)</p> <p>Arithmetic. Positional number systems. Arithmetic in the binary system. Application: representation of numbers on a computer. Fixed-point numbers: sign-module, U1, U2, BCD notation. Floating-point numbers.</p> <p>Elements of logic. Propositional calculus: introduction to predicate calculus. Introduction of concepts: equivalent propositions, necessary/sufficient condition, tautology, etc. Applications: logic gates or (in theory) programming in logic.</p> <p>Combinatorics. Examples of methods for counting elements of sets (Pascal's triangle, Venn diagram). Concepts: variation without repetition, variation with repetition, permutation, combination without repetition, combination with repetition, power set. Applications: lotto, poker, sports competitions.</p> <p>Graphs. Introduction to the concepts of directed/undirected graphs, trees, cycles in graphs, connected graphs, weighted graphs, etc. Trees with a distinguished root (motivation: data structures in computer science). Binary trees: labelling, tree traversal (pre-order, in-order, post-order algorithms). Graph search (breadth-first, depth-first). Binary search tree. Applications: examples of graphs, e.g. to find the shortest path, find an element, insert an element.</p> |
| Fundamentals of law in computer science | <p>INF_I_inž_W11, INF_I_inž_W12, INF_I_inž_W16, INF_I_inž_U01, INF_I_inž_U08, INF_I_inž_K02</p> | <p>Elements of constitutional law Elements of civil law Elements of criminal law Elements of administrative law Elements of commercial law Elements of labour law Intellectual property of computer scientists in the light of copyright and industrial property law</p> |
| Information technology | <p>INF_I_inž_W13, INF_I_inž_U05, INF_I_inž_U18 INF_I_inž_K01</p> | <p>Acquisition and preparation of data for analysis using a spreadsheet - acquisition of external data from various sources, organising and formatting data, checking and applying data correctness</p> <p>Creating forms in a spreadsheet - automating data processing using controls and macros Acquiring data and information using survey forms</p> <p>Using graphic tools, basics of application mock-up design</p> <p>Presentation of data and information, e.g. via a website website</p> |

| | | |
|---------------------------------|---|--|
| <p>Programming fundamentals</p> | <p>INF_I_inz_W02, INF_I_inz_U11, INF_I_inz_U12, INF_I_inz_K01, INF_I_inz_K04</p> | <p>Algorithms, ways of writing algorithms, Visual Studio programming environment, program structure in C#. Variable types, variable declaration, constants, operators, literals. Conditional statements: if, if-else, exceptions. Iterative loops: for, while, do-while. Control statements: switch statement, jump statements: goto, break, continue. Creating arrays, multidimensional arrays, array initialisation, array properties, using arrays with a foreach loop. Array operations. Setting elements to a given value, copying arrays, reversing Method definition, method call, value returned by a method, shared and local variables, exception reporting in a method. Declaring and calling methods with parameters, ways of passing arguments to a method, reference type arguments, passing any number of arguments, method name overloading, Main method argument. The concept of recursion, example - Hanoi Towers, abuse of recursion, algorithms with returns, creating function libraries. Structures in C#, FIFO queue, LIFO queue - stack. Unidirectional list: adding, removing an element from a list, other types of lists. Binary search tree: adding, removing, searching.</p> |
| <p>Management - Foundations</p> | <p>INF_I_inz_W12, INF_I_Eng_W16, INF_I_Eng_U02, INF_I_Eng_U05, INF_I_Eng_U08, INF_I_Eng_U12 INF_I_Eng_K01, INF_I_Eng_K03, INF_I_Eng_K05</p> | <p>Organisations – concept, models, characteristics, typology, objectives and measures of organisational effectiveness. Management – concept, forms of effective action, stages of the management process, management principles, change management (process, causes and overcoming resistance) Manager – concept, classification, characteristics, roles and competences of managers, authority, management styles, leadership (essence, characteristics, types, leadership of the future) Planning in an organisation – concept and essence of planning, stages of the planning process, types and hierarchy of plans, principles of planning, barriers to the effectiveness of plans The impact of the environment on the functioning of an organisation Decision-making – essence, identification of opportunities and threats, decision-making in conditions of uncertainty (concept of risk), types of decisions, decision-making model, game theory Organising – concept, components, organisational structures (types and characteristics), organisational procedures, formalisation of behaviours and work processes Motivation – assumptions about motivation and motivating, theories and views on motivation, tools and principles of motivation, incentive systems Organisational culture – concept, elements, levels (E. Schein, J. Kotter and J. Heskett), classification (flexible and rigid cultures), relationship between organisational culture and strategy and effectiveness</p> |
| <p>Introduction to studying</p> | <p>INF_I_inz_W04, INF_I_inz_U03, INF_I_inz_K01</p> | <p>University structure - Authorities, Dean's Office, Library, Career Office, Financial Services Department, International Relations Department Student organisations - Activities of the Student Council, Scholarship system , Individual organisation of studies</p> |

| | | |
|--------------------------------|---|---|
| | | Computerisation of studies - student accounts, electronic correspondence , Moodle platform, Extranet |
| Linear algebra with geometry | INF_I_inž_W01, INF_I_inž_U09 INF_I_inž_K01 | <p>Matrices: basic concepts, algebraic operations on matrices. Determinants: definition, properties and calculation of determinants - Sarrus' method, Laplace expansion using elementary transformations. Applications of determinants: singular and invertible matrices, inverse matrices and their determination and properties, Cramer's method for solving systems of equations, matrix equations, order of matrices, Kronecker-Capelli theorem, solving linear systems of equations using Gauss elimination.</p> <p>Linear spaces: linear combination of vectors, linear independence, basis, dimension, linear subspace. Scalar product, vector length, vector orthogonality, orthogonal and orthonormal bases.</p> <p>Vectors and eigenvalues of linear mappings.</p> <p>Examples of linear transformations and their matrices on a plane – translation, scaling, rotation relative to the origin of the coordinate system, Composition of transformations – scaling and rotation relative to any point, Homogeneous coordinates – transformation matrices on the plane in homogeneous coordinates – translation, scaling, rotation relative to the origin of the coordinate system, rotation relative to any point.</p> <p>Complex numbers. Geometric interpretation of complex numbers. Addition and multiplication of complex numbers and geometric interpretation , Exponentiation of complex numbers – De Moivre's formula,</p> |
| Computer systems architectures | INF_I_inž_W07 INF_I_inž_U09 INF_I_inž_U20, INF_I_inž_K01 | <p>The structure of a typical computer motherboard. The layout of integrated circuits on a typical motherboard.</p> <p>Introduction to computer system architecture and computer system structure. Programmable interrupt and DMA systems, memory systems, communication interfaces.</p> <p>Processor architecture, cache memory, processor software model, system buses, families of popular microcontrollers. Arduino and Raspberry Pi systems.</p> <p>Integration of assembler and C++ languages. Assembler inserts in Visual Studio, their execution and integration with the rest of the code.</p> <p>Basic COM port (RS232) support, port configuration, sending characters, checking the number of characters waiting in the buffer, reading characters from the buffer.</p> <p>Controlling the robot via a radio module with a USB interface seen as a COM port, introducing joystick support in Windows. Using microcontroller runtime tools along with learning about the software environment and hardware tools, as well as methods of running and inspecting programme code along with searching for hardware design errors.</p> <p>Support for LSD displays, sensors and other actuators, along with programming of microcontroller hardware resources</p> <p>ARM9 with Windows CE using touchpad events on the screen, timing circuits and methods of drawing on the screen.</p> |
| Methods of writing projects | INF_I_inž_W09, INF_I_inž_W12, INF_I_inž_U01, INF_I_inž_U03, INF_I_inž_K01, INF_I_inž_K04, INF_I_inž_K05 | <p>Familiarisation with the principles of project and written work and the correct creation of documentation.</p> <p>Stages of project work Literature review</p> <p>Creation of a research tool</p> <p>Creating documentation</p> |

| | | |
|---|--|---|
| Electronics fundamentals | INF_I_Eng_W01, INF_I_Eng_U04, INF_I_Eng_U08, INF_I_Eng_K02 | Electric current – physical fundamentals (direct and alternating current) Laws of electrical circuits Analysis of simple electrical circuits Semiconductors (diodes, bipolar transistors) Cells and batteries Investigation of electrical circuit laws Investigation of semiconductor operation Construction of simple digital circuits Construction of operational amplifiers |
| Network technologies | INF_I_inž_W05, INF_I_inž_U16 INF_I_inž_K01 | Network reference models, network types, computer network topologies. Types of network devices. Classification of computer networks. General principles of network addressing. IP address classes. Detailed rules for configuring IP addressing. Division of networks into subnets. Designing and configuring network topologies. Designing and configuring basic network services. Designing and configuring network protocols. Computer network design (case study). |
| Object-oriented programming | INF_I_inž_W02 INF_I_inž_U09 INF_I_inž_U13, INF_I_inž_U14, INF_I_inž_K01 | Defining classes and objects in C#. Discussion of access modifiers. Comparison of the properties of structures and classes. Object constructors and destructors, initialising structures. Definition of properties Static components: fields, methods, constructors, and classes. Operator overloading. Inheritance and method overriding. Reflection mechanism and defining additional information in the form of code attributes. Class delegation mechanism. Interface design. Interface inheritance. Class diagram design. Creating UML diagrams Object-oriented programme design Object-oriented programme implementation Creating object-oriented programme documentation |
| Systems engineering and system analysis | INF_I_inž_W07 INF_I_inž_U09 INF_I_inž_U22 INF_I_inž_K03 | Basic concepts in systems theory System structure, system hierarchy and basic concepts system concepts Life cycle of natural and artificial systems. System design methods. Designing information systems and using modern IT tools for system design System modelling, simulation and optimisation Process design basics and process management concepts Process identification, mapping and sample process design software Process performance measurement and methods for improving processes Project design (e.g. for a company) including: project characteristics, needs assessment, project vision, contextual model, information and physical flow diagrams, system function diagrams, project implementation costs, project financing, project implementation schedule Project analysis (e.g. for a company) including: identification and description of processes, process maps |
| Statistics | INF_I_inž_W01, INF_I_inž_W12, INF_I_inž_U09 INF_I_inž_K01, INF_I_inž_K02 | Probability (definition and properties of probability, conditional probability, independence of events, formula for total probability, Bayes' formula, examples of applications in computer science). Probability theory concepts useful in statistical inference (concepts related to normal and two-point distribution, point estimators). Selected discrete distributions: |

| | | |
|--|--|---|
| | | <p>bipartite, binomial, geometric, Poisson; examples of applications in computer science). Descriptive analysis of the structure of phenomena. Measuring the concentration of a phenomenon and comparing the structures of phenomena Time series analysis. Individual and aggregate indicators of dynamics. Development trends of phenomena - trends and their types. Graphical presentation of data and its use Random variable and its distribution. Parameters of the distribution of a random variable Point and interval estimation Confidence interval and level. Testing statistical hypotheses Analysis of variance, correlation and regression</p> |
| Communication and presentation - workshops | INF_I_inž_W12, INF_I_inž_U06, INF_I_inž_K04 | <p>Introduction to the subject. Fundamentals of communication - concepts and definitions. Verbal and non-verbal communication. Principles of giving feedback. Social influence in communication. The speaker in public speaking - self-presentation. Ways of coping with stress. Basic principles of creating multimedia presentations using modern technologies.</p> |
| Databases | INF_I_inž_W02 INF_I_inž_W04, INF_I_inž_U10_in ž, INF_I_inž_U15, INF_I_inž_U19_in ž, INF_I_inž_K01 | <p>The concept of a database and database management system. Entity relationship model. Relationships between entities. Entity relationship diagrams. Physical organisation of databases. Client-server architecture. Relational databases. Relational algebra. Relational calculus. Relational query languages. Relational database design. Functional dependencies. Normal forms. SQL language. Introduction to SQL. Advanced SQL features. Protecting databases against misuse (perspectives, data integrity). Concurrent database operations. Locks. Transactions. Stored procedures. Triggered procedures. Accessing the database from external applications. Programming interfaces.</p> |
| Operations research | INF_I_inž_W06, INF_I_inž_W12, INF_I_inž_U05, INF_I_inž_U07, INF_I_inž_U22_ INF_I_inž_K01 | <p>Linear programming 1. Examples of linear programming models - optimal production structure model, diet model, mixture model, technological process selection problem. 2. Formulation of linear programming models based on content tasks. 3. Graphical solution. 4. Sensitivity analysis - Sensitivity analysis of the optimal solution to changes in the coefficients of the objective function. Sensitivity analysis of the optimal solution to changes in free terms. 5. Use of the SOLVER add-in Transportation problem 1. Open and closed problems 2. Transportation and production problems 3. Assignment problem 4. Travelling salesman problem 5. Use of the SOLVER add-in Network programming 1. Network analysis (CPM-Critical Path Method) 2. Project time-cost analysis (CPM-COST), cost gradient, minimum project duration 3. Minimisation of project implementation cost at a given time 4. Minimising project implementation time for a given cost 5. PERT method – expected project implementation time and its variance, probability of project implementation within a given time, project implementation time with a given probability 6. Use of the SOLVER add-on Non-linear programming 1. Lagrange multiplier method 2. Use of SOLVER in non-linear programming</p> |

| | | |
|--|---|---|
| Foreign language | INF_I_Eng_W11, INF_I_Eng_W12, INF_I_Eng_W15, INF_I_Eng_U03, INF_I_Eng_U04, INF_I_Eng_K01, INF_I_Eng_K04 | Types of computers/inside the computer Spreadsheets: uses, actions, formatting, Desktop publishing Image and design Web design, web browsing Video conferencing Social media Operating systems Networking Viruses, anti-virus software, firewalls Careers in computing Business Talks |
| Algorithms and data structures | INF_I_Eng_W03, INF_I_Eng_U04, INF_I_Eng_U12, INF_I_Eng_K01, INF_I_Eng_K05 | Introduction. Abstract Data Structures as a combination of algorithms and data structures. Fundamentals of computational complexity. Array as an example of ASD. Encapsulation, interface, implementation. Implementations: basic, extensible, sparse. List. List interface. Types of implementations. Single and double-linked lists with references. Array-based lists. Ordered and sorted lists. Recommendations for using lists. Stack and queue - structures with destructive reading. Stack and queue interface. Implementations. Circular queue. Priority queue. Structures with mixing functions. Characteristics of a mixing function. Set – array-based implementation. Size change. Two-level implementation. Map. Recommendations for use. Trees and heaps. Terminology. Trees with pointers: operations. Tree traversal. Balanced trees: operations. Heaps as lists and as arrays. Sorting arrays. Features of sorting algorithms. Implementations: bubble sort, quick sort, heap sort, merge sort. Comparative analysis of algorithms and their complexity. Computational complexity. Computational models. Types and ranges of complexity. Typical values. P, NP, undecidable, uncomputable problems. Parallelisation possibilities Graph. Terminology. Examples of use. Interface. Implementation with a matrix and a list. Visiting vertices. Cycle detection. Shortest path. Minimum spanning tree. Topological sorting. Graph colouring. Travelling salesman problem. Algorithms. Recursion and iteration. Brute force, greedy, divide and conquer, dynamic, heuristic algorithms in applications. Concurrency problems. Terminology. Deadlocks and races. Challenges: thread-safe, thread-useful, thread-efficient. Examples of synchronisation and isolation. |
| Modelling and analysis of computer science systems | INF_I_inž_W10_in ž, INF_I_inž_W14, INF_I_inž_U04, INF_I_inž_U08, INF_I_inž_U11_in ž, INF_I_inž_U14, INF_I_inž_K01 | Introduction. Terminology. System, environment, domain. Scope of the lecture. Life cycle. The role of analysis and modelling in the software life cycle. Historical overview. Role in an agile organisation. Feasibility study. Requirements. The role and importance of requirements in analysis and modelling. Requirements gathering in the development process. The 'epic' process, user stories, notations. Methods of requirements gathering. Use of UML and SysML. Architectural styles. The role of architectural styles in analysis and modelling. Classification of styles with a focus on selected styles. Examples. Reverse engineering as a method of system analysis. Acquiring knowledge from code. Automatic and assisted methods. Tools and examples. |

| | | |
|---|--|--|
| | | <p>Object-oriented analysis. The importance of the paradigm in analysis. Contemporary paradigms. Concepts of object-oriented analysis: class-object, relation.</p> <p>Acquiring elements of analysis. Examples</p> <p>System model. Model categories. Modelling levels and perspectives. Characteristics of a good model. Ways of describing a model</p> <p>Analysis and modelling in practice: an extensive example. Transition from a real design problem to a detailed model using approaches and techniques from the lecture. (W:1)</p> <p>Tools, workshops and environments. Integrated tools. Characteristics of selected tools.</p> |
| Operating Systems | <p>INF_I_inž_W03, INF_I_inž_W07_in ž, INF_I_inž_W08, INF_I_inž_U11_in ,</p> | <p>Operating system architecture</p> <p>Challenges facing operating system designers</p> <p>Single-threaded and multi-threaded architecture based on architecture</p> <p>Intel multicore architecture</p> <p>Monitoring, diagnosing and managing RAM and processor.</p> <p>The concept of cache at different levels of the operating system and hardware, including a discussion of the current memory hierarchy and significant changes resulting from the emergence of Intel Optane technology.</p> <p>Overview of the types, capabilities, advantages and disadvantages of virtual environments including four models of operating system virtualisation.</p> <p>Detailed discussion of the concept of filesystem virtualisation – containerisation using Docker as an example.</p> <p>Installation of operating systems.</p> <p>Management of Docker instances through management of local resources and using the Docker Hub repository</p> <p>Extension of Docker with Kubernetes.</p> |
| Digital and microprocessor technology with IoT elements | <p>INF_I_inž_W02_in ž, INF_I_inž_W07_in ž, INF_I_inž_W09, INF_I_inž_U01, INF_I_inž_U04, INF_I_inž_U10_in ž, INF_I_inž_U13, INF_I_inž_K01</p> | <p>Introduction to digital technology, number systems with different bases, methods of converting numbers between different notations (at least binary and hexadecimal systems), Boolean algebra (basic theorems, De Morgan's laws), logic gates, implementation of simple functions, Karnaugh's method, examples of applications.</p> <p>Introduction to microprocessor technology, discussion of modern microcontrollers with particular emphasis on the types of microcontrollers available in the laboratory.</p> <p>Discussion of the Arduino IDE and ecosystem (characteristics of the hardware layer of modules and extensions, programming environment, compiling programmes, use of a programmer/loader, serial port monitor, modules with microcontrollers, components and subassemblies available in the laboratory). Arduino programming – overview/review of the basic syntax of C/C++, taking into account the specifics of Arduino, demonstration and discussion of program examples and standard libraries.</p> <p>I/O devices, peripheral circuits. Analogue-to-digital and digital-to-analogue conversion. Measurement of physical quantities and control of actuators. User interface - buttons, knobs (potentiometers, encoders), keyboard, alphanumeric, graphic and touch displays.</p> <p>Elements of "classical" digital technology (in combination with microprocessor technology) – discussion and demonstration with exercises on digital circuits (counters, I/O buffers, multiplexers, demultiplexers, etc.)</p> <p>Advanced Arduino programming – Atmel Studio environment, overview of low-level programming elements, hardware registers, hardware and software interrupt handling, I/O devices without using Arduino library functions.</p> |

| | | |
|--|--|---|
| Scripting languages programming | INF_I_Eng_W02_Eng, INF_I_Eng_W02, INF_I_Eng_U10_Eng, INF_I_Eng_U11_Eng, INF_I_Eng_K03 | Introduction to scripting languages Simple and complex data types Program flow control Own functions and built-in functions Generators Modules and packages Exceptions Numerical calculations File operations |
| Designing information systems | INF_I_inž_W07_in ž, INF_I_inž_W10_in ž, INF_I_inž_U13, INF_I_inž_U22_in ž, INF_I_inž_K02 | Information system architectures (revision). Introduction to structural design. Interface design. Form and report design processes. Designing dialogue boxes and dialogue sequences The use of guides and checklists in design processes Structuring and controlling input data Prototyping in system design (prototype construction and verification) Introduction to object-oriented design. Building UML diagrams (use cases, activities, classes, relationships between diagrams). IT tools for system design Use case diagrams – PU identification, advanced relationship specification, introducing stereotypes into the model, managing the complexity of extensive use case models using packages Modelling interactions in an IT system – conceptual, implementation and occurrence sequence diagrams, UC specification using implementation sequence diagrams Activity diagrams and class diagrams – PU specification, data modelling techniques in activity diagrams, activity diagram extensions for real-time system modelling Class diagrams – modelling data structure in the system, implementing the data model, exemplifying data structure using object diagrams State machine diagrams – tracking object states in the system, nesting state machines, pseudo-states Modelling hardware infrastructure and embedding software components using component diagrams and UML deployment diagrams |
| Modelling and computer simulations | INF_I_Eng_W01, INF_I_Eng_U09_Eng, INF_I_Eng_K01 | Objectives, application, stages of modelling. Errors in numerical calculations. The Scilab package as an example of a modelling and simulation tool: programme structure, entering commands, executing commands, using the help system, calculation accuracy, mathematical constants, numbers and variables, lists, vectors and matrices. Graphics: 2D and 3D graphs, animation, graphics export Differential schemes for solving ordinary differential equations Physical model of the dynamics of material point systems Modelling examples (percolation, random walk) |
| Visualisation and exploration of business data | INF_I_inž_W06, INF_I_inž_U07, INF_I_inž_K04 | The importance of business data. Areas of application of business data analysis and visualisation. Development of methods for exploring and visualising business data |

| | | |
|--------------------|---|--|
| | | <p>Data sources. Acquiring data for analysis, preparing data for analysis</p> <p>Data analysis and visualisation tools - classification and characteristics Principles and forms of data visualisation</p> <p>Application of selected tools in the data analysis process in the following areas: -</p> <p>Preparing data for analysis - Data analysis and visualisation - Presentation of business data. Building dashboards, stories and reports</p> <p>Sharing business reports</p> |
| Internship | <p>INF_I_inž_W04, INF_I_inž_W16, INF_I_inž_W17,</p> | <p>Familiarising students with the duties of a position in a selected business entity and performing actual tasks assigned by the internship supervisor.</p> <p>Familiarising students with the health and safety regulations in force at the company. Legal basis and scope of the institution's activities. Legal status, ownership structure, scope and range of the institution's activities. Mission and strategy of the institution. Organisation of the business entity and the scope of its activities. Organisational structure, decision-making powers and responsibilities of individual units. Organisational documentation of the institution. Statutes, regulations, instructions and rules for document circulation.</p> <p>Familiarising the student with technical documentation related to the internship programme.</p> <p>Familiarising students with the equipment and IT tools used in the company.</p> <p>Familiarisation with the methods of work in the company.</p> <p>Familiarisation with the practical application of IT systems.</p> <p>Practical implementation of tasks assigned to the student in the company.</p> |
| Advanced databases | <p>INF_I_inž_W02_in ž, INF_I_inž_W04, INF_I_inž_U15, INF_I_inž_U20, INF_I_inž_K03</p> | <p>Installing Oracle Database 11g Express Edition, designing databases in Oracle SQL Developer Data Modeler</p> <p>SQL syntax refresher (SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, DROP, GRANT, REVOKE)</p> <p>Executing SQL and PL/SQL queries using Oracle SQL Developer User permissions in Oracle – users, system permissions, object permissions, predefined roles, user-defined roles, profiles</p> <p>Basic elements of the PL/SQL language - blocks, block structure, sections, delimiters, identifiers, literals, data types.</p> <p>Variables, variable scope, related variables, conditional statements (IF, CASE with expression and search) in PL/SQL</p> <p>Iterative statements (LOOP, FOR and WHILE), loop naming, labels in loops, GOTO and CONTINUE statements, data retrieval (INSERT ... INTO ...), records - explicit and implicit definitions</p> <p>Types of cursors – declaration, opening, data retrieval, closing, cursor attributes, cursors with parameters, cursors and LOOP, WHILE and FOR loops, FOR UPDATE clause, implicit cursors, exceptions in PL/SQL Procedures without parameters and with parameters, IN, OUT and IN OUT modes, functions without parameters and with parameters, recursive functions and functions in SQL queries, NOCOPY modifier in PL/SQL</p> <p>BEFORE, AFTER and INSTEAD OF triggers, :old and :new identifiers, REFERENCING and WHEN clauses, sequences and triggers in PL/SQL</p> <p>Packages in Oracle - package specification and body, overloading subprograms from packages, packages for serial execution, initialising packages, built-in packages (DBMS_OUTPUT, UTL_FILE)</p> |

| | | |
|---|--|--|
| | | <p>Collections in Oracle - associative arrays, VARRAY arrays, nested arrays.</p> <p>Object types, object tables, object table attributes, object table duality, references and dereferences. Creating a database interface in APEX (Oracle Application Express)</p> |
| Commercialisation project management | <p>INF_I_inž_W04, INF_I_inž_W15, INF_I_inž_U01, INF_I_inž_U17_in ž, INF_I_inž_U18_in ž, INF_I_inž_U19_in ž, INF_I_inž_U20, INF_I_inž_K05</p> | <p>The scope and tasks set for students as part of the research and implementation project depend on the specific nature of the problem, which will be proposed by companies cooperating with the Edge Computing specialisation. At the beginning of the semester, students will be familiarised with the design challenges proposed by companies (5-8 topics). The topics of the projects will be related, among others, to software running on specific hardware, e.g. detection of people and objects in camera images, recognition of road signs in videos, mobile systems for diagnosis, reporting, authentication, etc. The projects will be advanced, with fairly extensive functionality, so that a group of approx. 5 students to be able to complete it</p> |
| Information security management - workshops | <p>INF_I_inž_W04, INF_I_inž_U20, INF_I_inž_K05</p> | <p>Basic standard defining the Information Security Management System Technological aspects of information security Implementation of information security management solutions Identification of risk management processes in the field of information security Generating a risk management procedure and conducting risk management for information security New challenges and monitoring the effectiveness of information security management Good practices in information security – people and technology ISMS model for a selected company</p> |
| General electives | <p>INF_I_inž_W09, INF_I_inž_W12, INF_I_inž_U02, INF_I_inž_U12, INF_I_inž_K05</p> | <p>These are subjects taught in English. Subjects: <i>Agile software development</i> (During the Agile software development course, students will learn about modern methodologies used in IT companies to develop and validate software products), <i>PLM - Product Lifecycle Management IT systems</i> (discussion of the most popular production systems that manage the product lifecycle, challenges and difficulties faced by the PLM support centre), <i>Application of selected Scrum Framework and SixSigma tools to optimise the support centre</i> (identification of Scrum and Six Sigma tools that may be most useful for the IT infrastructure support centre), <i>Applied Statistics with R</i> (Aim of the course is to introduce the students to quantitative methods for analysing data coming from different sources), <i>Management of IOT Open Data Projects in Smart Cities, Startups and scaleups in the global tech ecosystem.</i></p> |
| Preparing the student for the labour market | <p>INF_I_inž_W16, INF_I_inž_U03, INF_I_inž_U08,</p> | <p>The labour market and changes taking place in it. What to look out for in order not to be deceived in the labour market. Determining your competences and professional preferences, analysing your strengths and weaknesses. Interpersonal skills in job hunting. Verbal and non-verbal communication. How to prepare for an interview with an employer. Business etiquette, or what is and is not appropriate in the workplace. Application documents, job search methods and techniques</p> |
| Commercialisation project | <p>INF_I_inž_W04, INF_I_inž_W15,</p> | <p>The essence of an implementation project, basic principles of project writing, project evaluation criteria;</p> |

| | | |
|-------------------------|--|--|
| | INF_I_inž_U01, INF_I_inž_U17_in ž, INF_I_inž_U18_in ž, INF_I_inž_U19_in ž, INF_I_inž_U20 | Selection of a company/research area for an implementation project. Project objective, methods of measuring variables and sources of information. Development and analysis of collected research material. Identification of opportunities and threats related to the implementation of the project. Determination and evaluation of possible project implementations. Formulation of conclusions from the project |
| Distributed processing | INF_I_inž_W04, INF_I_inž_W05 | WebClient class Thread class Task class ThreadPool class BackgroundWorker class Cloud environments Microsoft Azure virtual machines and their configuration |
| Artificial intelligence | INF_I_inž_W06, INF_I_inž_U17_in ž, INF_I_inž_K02 | Knowledge representation methods Genetic algorithms Rule-based systems: representation and inference Systems with uncertain knowledge: use of fuzzy logic Machine learning, artificial neural networks |
| ICT network threats | INF_I_Eng_W06, INF_I_Eng_U08, INF_I_Eng_U16_Eng, INF_I_Eng_K01 | Configuring network devices using SSH and automatic security features. Configuring local AAA authentication. Configuring server-based authentication using the RADIUS service. Hashing a text file using OpenSSL. Encrypting and decrypting data using OpenSSL and a hacking tool. Examining Telnet and SSH in Wireshark. Configuring site-to-site VPNs. Configuring basic ASA network settings and services, routing, and DMZ using ACLs. Configuring a firewall. |
| Physical education | INF_I_inž_W17, INF_I_inž_U08, INF_I_inž_U09_in ž, INF_I_inž_K02 | Discussion of the rules for the safe use of facilities, equipment and environments related to various sports. Familiarisation with the rules and regulations of the sports hall. Organisation, hygiene and order at work. Discussion of the rules and criteria for passing the course. Exercises for developing correct posture using equipment and accessories. Basic exercises for specific muscle groups. Training to develop abdominal muscles; Training to develop chest muscles; Training to develop leg muscles; Training to develop arm muscles; Training to develop shoulder girdle muscles; Training to develop back muscles; Training to develop calf muscles Team games: rules and regulations of team games, team game techniques and tactics Self-defence elements: defence against punches and kicks; defence against grabs; defence against knife threats; defence against firearm threats The educational value of sport The importance of physical activity in human life |
| Programming in Java | INF_I_inž_W04, INF_I_inž_W16, INF_I_inž_W17, | Conditional statements, Loops, Methods, ArrayList Object-oriented programming, Arrays, Sorting, Control statements, Searching, HashMap Interfaces, Exceptions, I/O, Inheritance |

| | | |
|---|--|---|
| | | Graphical user interface (GUI), classes and objects, use of library classes Regular expressions, Iterators, Enum |
| .NET programming | INF_I_inž_W02_in ž, INF_I_inž_W04, INF_I_inž_U15, INF_I_inž_U20, INF_I_inž_K03 | Architecture of the .NET platform, Framework and related technologies. Streams of access to file data. Data serialisation. Monitoring application performance using system counters. Tracking application performance by logging events in the system. Data encryption. Interaction with Windows API. Multithreaded programming. Integration unmanaged code |
| Advanced Internet technologies | INF_I_inž_W04, INF_I_inž_W15, INF_I_inž_U20, INF_I_inž_K05 | HTML 5, New semantic elements in HTML, HTML5 forms and their validation, Geolocation and interactive map with Google Maps, Web Storage and JSON data storage Rest api. Overview of basic specifications. Typical solutions, tools for working with data. Node.JS, Installation, launch and configuration of the engine. Basic commands, writing functions. Creating your own node.js web server, Routing, express framework PWA, Service Workers – registration, installation, activation. Best practices for code preparation. |
| Mobile devices programming | INF_I_inž_W09, INF_I_inž_W12, INF_I_inž_U02, INF_I_inž_U12, INF_I_inž_K05 | Determinants and approaches to programming multi-platform mobile applications Architecture of cross-platform mobile application design platforms in jQuery Mobile and Ionic. Designing a web mobile application in the jQuery Mobile framework: Pages Patterns Menus Transitions Columns Interactive panels Interactive lists Forms: templates, controls, interactive panels. Designing a multi-platform mobile application in the Ionic framework: Creating a multi-platform application project Basic project components Defining the user interface Modifying the application's appearance styles Designing the data access layer Communication with the data server Native functions for mobile devices: vibration, camera, video camera Compiling and implementing the application on a mobile device |
| Programming and testing of web applications | INF_I_inž_W16, INF_I_inž_U03, INF_I_inž_U08, | Fetch API / Ajax (XHR). ES6+, Javascript modularity, TypeScript, Unit, integration and functional testing of web applications Mocha, Chai, Ava, Jasmine |
| Advanced object-oriented programming | INF_I_inž_W04, INF_I_inž_U01, INF_I_inž_U17_in ž, INF_I_inž_U18_in ž, INF_I_inž_U19_in ž, INF_I_inž_U20, INF_I_inž_K05 | Defining overload operators (mathematical, logical, conversion) Defining and handling events. Defining template classes Defining and implementing template classes Building and enumerating non-standard class collections. LINQ language for data processing. Defining asynchrony. Design patterns |
| Functional programming | INF_I_inž_W04, INF_I_inž_W05, | Definition of functional programming. Haskell language, overview of tools and compilers The concept of functions, defining functions and function types. Currying and partial argument assignment. Function composition. Basic data types. Defining modules. Lambda calculus Pattern matching. Type inference Recursion. Lazy evaluation. Higher-order functions |
| Embedded systems programming - team project | INF_I_inž_W06, INF_I_inž_U17_in ž, INF_I_inž_K02 | Introduction to embedded systems, including definitions, applications, characteristics and history of embedded systems. Architecture of embedded systems, description, characteristics and measurements of input/output devices and AVR circuits |

| | | |
|--|--|---|
| | | <p>Digital signals and methods of their processing, CPS Application of the appropriate programming language for microcontroller programming. Real-time operating systems and their application. Network protocols and interfaces in embedded systems. Communication models Design of control and measurement systems. Examples of embedded system applications</p> |
|--|--|---|

IV. ASSIGNMENT OF FIELD OF STUDY TO SCIENTIFIC DISCIPLINES

| No. | Scientific disciplines | % OF ECTS POINTS |
|-----|--|------------------|
| 1. | Computer Science (leading discipline) | 57 |
| 2 | Technical Informatics and Telecommunications | 33 |
| 3 | Mathematics | 10 |

V. BASIC ECTS INDICATORS SPECIFIED FOR THE STUDY PROGRAMME

| Indicator name | Number of ECTS credits/Number of hours |
|---|--|
| Total number of ECTS credits that a student must obtain as part of classes conducted with the direct participation of academic teachers or other persons conducting classes | 113.1 |
| Total number points ECTS assigned to classes developing practical skills | 153.6 |
| Total number of ECTS points that a student must obtain in classes in the field of humanities or social sciences – in the case of fields of study assigned to disciplines in fields other than humanities or social sciences, respectively | 5 |
| Total number of ECTS points assigned to elective classes | 87.4 |
| Total number points ECTS assigned to 39 | 39 |

VI. SCOPE, RULES AND FORMS OF PROFESSIONAL INTERNSHIPS

Duration: 960 hours of professional practice.

The student must complete the designated number of hours (in total) by the end of the programme.

Students can organise their work placements individually or with the help of the Student Services Office.

Students going on work placements should complete a preliminary statement, on the basis of which a contract and referral (for the employer) are drawn up, as well as a certificate of completion, work placement programme, work placement time card and learning outcomes. After completing the internship, the student submits all documentation to the university, where it is reviewed by the internship supervisor.

Internships can take place in a stationary or hybrid form.

a. METHODS OF VERIFYING AND ASSESSING THE LEARNING OUTCOMES ACHIEVED BY THE STUDENT THROUGHOUT THE ENTIRE EDUCATION CYCLE FOR A GIVEN FIELD OF STUDY

| Verification method/Areas | Knowledge | Skills | Social Social |
|---|-----------|--------|------------------|
| Test, written examination | X | X | |
| Written examination | X | X | |
| Oral examination | X | X | |
| Project | X | X | X |
| Department | X | X | |
| Presentation | X | X | X |
| Essay | X | X | X |
| Individual oral presentation | X | X | |
| Participation in discussion | | X | X |
| Case study | X | X | |
| Report, field task report | X | X | |
| Practical tasks | | X | X |
| Independent task solving | | X | |
| Active participation in classes | | X | X |
| Observation of students by the academic teacher academic | | | X |
| Diploma examination | X | X | X |

The forms and methods of conducting classes, as well as the assessment criteria and its components for individual classes, are specified in the course description.