ERASMUS+ INCOMING STUDENTS
ACADEMIC OFFER - COURSES TAUGHT IN ENGLISH
COMPUTER SCIENCE
WINTER 2024/2025

1st year, 1 st semester

COMPUTER SCIENCE
BACHELOR

Mathematical analysis and linear algebra – lecture

The aim of the course is to present the concept of partial derivative and derivative of functions of several variables and the methods of their determination and application. The students will get acquainted with the concepts of the basics of mathematical analysis and with their properties, and learn the concept of a sequence limit and the techniques of calculating limits, the concept of a limit and a function along with its properties and calculation techniques, the definition of the derivative of a function of one variable and theorems concerning the calculus of derivatives and their applications for optimization purposes.

The aims of the subject:

- Presentation of the concept of partial derivatives and derivatives of functions of multiple variables, along with methods for determining and applying them.
- Introduction to the basic concepts of mathematical analysis and their properties.
- Understanding the concept of sequence limits and techniques for calculating limits, the concept of function limits and their properties, techniques for calculating derivatives of single-variable functions, and theorems concerning differential calculus and its applications in optimization.

in the the terms of knowledge

- Knows methods for calculating the limit of a sequence and functions and can use them effectively.

- Lists methods for calculating derivatives of functions (including inverse functions).
- Analyzes methods for using differential calculus to construct function graphs and calculate limits.
- Identifies methods for determining the antiderivative of a given function and methods for calculating definite integrals.
- Lists methods for performing operations with complex numbers, powers, and roots of numbers.

In the terms of skills

- Can calculate sequence and function limits using learned methods and use them effectively.
- Applies learned methods to calculate derivatives of functions, including composite and inverse functions.
- Uses differential calculus to solve optimization problems.
- Determines antiderivatives for a given integrable function and finds definite integrals.
- Performs operations with complex numbers, calculates roots and powers for such numbers, and solves simple equations in the complex domain.
- Performs operations with matrices, calculates determinants of square matrices, examines their ranks, inverts non-singular matrices, and solves general systems of linear equations.

In the terms of social competencies

- Possesses the ability to tackle practical problems based on the acquired knowledge and skills.
- Can make accurate self-assessments of the actions performed.

Program Content

- Concept of sequence limit and its properties.
- Concept of function limit and function continuity.
- Basic theorems on function limits and properties of continuous functions.
- Concept of the derivative and its properties.
- Theorems on operations with derivatives, the derivative of composite functions, and the derivative of inverse functions.
- Necessary and sufficient conditions for the existence of local extrema of a single-variable function.
- Geometric properties of single-variable function derivatives.
- Partial derivative of functions of multiple variables, derivative of functions of multiple variables.
- Necessary and sufficient conditions for the existence of extrema of functions of multiple variables.
- Complex numbers and operations with them.
- Exponentiation and root extraction of complex numbers.
- Matrices and operations with matrices.
- Rank of matrices and determinant of square matrices.

- General systems of linear equations and theorems on solving them.

Learning Methods:

- Problem-based lecture
- Exercises
- Classic problem method
- Case study

Education verification method: exam, activity during classes

ZDW: Theoretical foundations of computer science / Information processing technologies

The aim of the course is to present the concept of partial derivative and derivative of functions of several variables and the methods of their determination and application. The students will get acquainted with the concepts of the basics of mathematical analysis and with their properties, and learn the concept of a sequence limit and the techniques of calculating limits, the concept of a limit and a function along with its properties and calculation techniques, the definition of the derivative of a function of one variable and theorems concerning the calculus of derivatives and their applications for optimization purposes.

The aims of the subject:

- Imparting knowledge in the subject area.
- Achieving the ability to practically apply computers and software in conducting and presenting socio-economic analyses.

Learning Outcomes

in the terms of knowledge

- Achieves knowledge in the subject area, elements of the computer system, and software typology.
- Knows and understands methods of analyzing the considered issue.
- Knows and understands the complex relationships between the analyzed data.

in the terms of skills

- Achieves the ability to apply computers and software in conducting and presenting socio-economic analyses.
- Can design and program uncomplicated IT tasks.
- Can use available literature sources to formulate and solve IT problems.

---

in the terms of social competences

- Demonstrates the ability for continuous education, improvement, and enhancement of professional, personal, and social competences.
- Understands the need for and knows the possibilities of continuously improving their competences.
- Is aware of the importance and understands the non-technical aspects and impacts of an IT engineer's activities.

Program Content

- Definition and subject of computer science. Areas of application of computer science. Data representation in computer memory. Information units: bit and byte.
- Structure of the computer system. Central processing unit. Operational memory. Definition and functions of the operating system. Peripheral devices.
- Definition and types of mass storage. Communication model with the computer. Input and output devices: audio/video, monitors, printers, plotters, scanners.
- Typology and applications of software: word processors, spreadsheets, presentation graphics, creating hypertext documents, Internet and global network services, email, web publishing.

Teaching Methods:

- Laboratory exercises
- Workshop method
- Didactic discussions

Education verification method: exam, activity during classes

ZDW: Theoretical foundations of computer science / Information processing technologies – tutorials

The aim of the course is to present the concept of partial derivative and derivative of functions of several variables and the methods of their determination and application. The students will get acquainted with the concepts of the basics of mathematical analysis and with their properties, and learn the concept of a sequence limit and the techniques of calculating limits, the concept of a limit and a function along with its properties and calculation techniques, the definition of the derivative of a function of one variable and theorems concerning the calculus of derivatives and their applications for optimization purposes.

The aims of the subject:

- Imparting knowledge in the subject area.
- Achieving the ability to practically apply computers and software in conducting and presenting socio-economic analyses.

Learning Outcomes

in the terms of knowledge

- Achieves knowledge in the subject area, elements of the computer system, and software typology.
- Knows and understands methods of analyzing the considered issue.
- Knows and understands the complex relationships between the analyzed data.

in the terms of skills

- Achieves the ability to apply computers and software in conducting and presenting socio-economic analyses.
- Can design and program uncomplicated IT tasks.
- Can use available literature sources to formulate and solve IT problems.

In the terms of social competencies

- Demonstrates the ability for continuous education, improvement, and enhancement of professional, personal, and social competences.
- Understands the need for and knows the possibilities of continuously improving their competences.
- Is aware of the importance and understands the non-technical aspects and impacts of an IT engineer's activities.

Program Content

- Definition and subject of computer science. Areas of application of computer science. Data representation in computer memory. Information units: bit and byte.
- Structure of the computer system. Central processing unit. Operational memory. Definition and functions of the operating system. Peripheral devices.
- Definition and types of mass storage. Communication model with the computer. Input and output devices: audio/video, monitors, printers, plotters, scanners.
- Typology and applications of software: word processors, spreadsheets, presentation graphics, creating hypertext documents, Internet and global network services, email, web publishing.

Teaching Methods:

- Laboratory exercises
- Workshop method
- Didactic discussions

Education verification method: exam, activity during classes

Algorithms and data structures – lecture

The aim of the course is to gain knowledge about the basics of the theory of algorithms (complexity, construction), simple and complex data structures. The students will acquire the ability to plan and construct complex algorithms, as well as learn about the possibility of choosing the optimal data structure for a given problem.

The Aims of the Subject

- Learning the basics of algorithm theory (complexity, construction), simple and complex data structures.
- Acquiring the skills to plan and construct complex algorithms.
- Understanding the options for choosing optimal data structures for a given problem.

Learning Outcomes

in the terms of knowledge

- Selects a mathematical method for the presented problem.
- Can design several solutions to a given problem.
- Can propose appropriate data structures.

in the terms of skills

- Conducts complexity analysis of an algorithm.
- Plans the stages of work in constructing complex algorithms.
- Creates a diagram for the given problem.
- Performs a comparative analysis of different algorithms solving the same problem.

in the terms of social competences

- Creatively develops solutions to unusual problems.
- Has the ability to work in teams.
- Ensures objective analysis and evaluation of alternative solutions to a given problem.

Program Content

- Basic concepts of algorithm theory.
- Recursion.
- Algorithm complexity analysis.
- Sorting algorithms.
- Searching algorithms.
- Numerical methods.
- Graph theory.
- Interpolation and approximation.
- Compact linear schemes.

Teaching Methods:

- Problem lecture
- Discussion
- Brainstorming
- Multimedia techniques

Education verification method: exam, activity during classes

Algorithms and data structures –tutorials

The aim of the course is to gain knowledge about the basics of the theory of algorithms (complexity, construction), simple and complex data structures. The students will acquire the ability to plan and construct complex algorithms, as well as learn about the possibility of choosing the optimal data structure for a

The Aims of the Subject

- Learning the basics of algorithm theory (complexity, construction), simple and complex data structures.
- Acquiring the skills to plan and construct complex algorithms.
- Understanding the options for choosing optimal data structures for a given problem.

Learning Outcomes

in the terms of knowledge

- Selects a mathematical method for the presented problem.
- Can design several solutions to a given problem.
- Can propose appropriate data structures.

in the terms of skills

- Conducts complexity analysis of an algorithm.
- Plans the stages of work in constructing complex algorithms.
- Creates a diagram for the given problem.
- Performs a comparative analysis of different algorithms solving the same problem.

In the terms of social competencies

- Creatively develops solutions to unusual problems.
- Has the ability to work in teams.
- Ensures objective analysis and evaluation of alternative solutions to a given problem.

Program Content

- Basic concepts of algorithm theory.
- Recursion.
- Algorithm complexity analysis.
- Sorting algorithms.

- Searching algorithms.
- Numerical methods.
- Graph theory.
- Interpolation and approximation.
- Compact linear schemes.

Teaching Methods:

- Problem lecture
- Discussion
- Brainstorming
- Multimedia techniques

Education verification method: exam, activity during classes

Occupational health and safety with elements of ergonomics – lectures

The aim of the course is to provide students with knowledge with knowledge in the field of occupational health and safety.

Learning Outcomes

in the terms of knowledge

- Knows the factors of the work environment.
- Knows, understands, and appropriately classifies work environment factors, including personal protective equipment and its purpose.
- Knows and understands work hygiene from the perspective of ergonomics and physiology.

in the terms of skills

1

- Makes accurate assessments of hazards present in the work environment.
- Can estimate occupational risks in various job positions.
- Selects personal protective equipment appropriate to the existing hazards.

in the terms of social competences

- Is responsible for workplace safety and hygiene.
- Has the ability to make decisions regarding the reduction of occupational risk.

Program Content

- The role of the subject.
- Workplace safety.
- Work hygiene.
- Ergonomics.

- Physiology.
- Classification of work environment factors: hazardous, harmful, and onerous.
- Types of factors: physical, chemical, biological.
- Analysis and assessment of work environment factors.
- Workplace accidents.
- Legal protection of labor.
- First aid (general issues).

Teaching Methods:

- Lecture
- Didactic discussion
- Analysis and interpretation of texts
- Workshop method

Education verification method: exam, activity during classes

Basics of programming 1 – lecture

The aim of the course is to gain knowledge in the field of the subject. The students will learn to design, design computer programs and create the correct source code, as well as run the created program code using the selected translator.

Learning Outcomes

in the terms of knowledge

- Achieves knowledge in the subject area, including building and developing computer programs.
- Knows and understands methods for analyzing the given problem.
- Knows and understands complex relationships between analyzed data.

in the terms of skills

- Achieves the ability to design correct source code for computer programs. Possesses skills in using selected compilers and tools for program development.
- Can design and program uncomplicated IT issues.
- Can use available literature to formulate and solve IT problems.

in the terms of social competences

- Understands the need and knows the possibilities for continuous improvement of professional, personal, and social qualifications.
- Understands the need and knows the possibilities for continuously enhancing their own competences.
- Is aware of the importance and understands non-technical aspects and

consequences of the work of a computer engineer.

Program Content

1. Definition, elements, and examples of programming languages. Language reports. High-level programming languages. Programming techniques – structured vs. object-oriented.
2. General structure of a computer program. Source and executable code. Translation methods. Compilation vs. interpretation of code.
3. Program construction. Data types. Constants, variables, arrays. Arithmetic and logical operators. Assignment instructions, conditional instructions: single and multiple selection.
4. Organizing repetition operations – iteration. Loop iteration counters. Increment vs. decrement. Variations of iterative instructions.

Teaching Methods:

- Lectures
- Project methods
- Auditory / laboratory exercises

Education verification method: exam, activity during classes

Basics of programming 1 – tutorials

The aim of the course is to gain knowledge in the field of the subject. The students will learn to design, design computer programs and create the correct source code, as well as run the created program code using the selected translator.

Learning Outcomes

in the terms of knowledge

- Achieves knowledge in the subject area, including building and developing computer programs.
- Knows and understands methods for analyzing the given problem.
- Knows and understands complex relationships between analyzed data.

in the terms of skills

- Achieves the ability to design correct source code for computer programs. Possesses skills in using selected compilers and tools for program development.
- Can design and program uncomplicated IT issues.
- Can use available literature to formulate and solve IT problems.

in the terms of social competences

- Understands the need and knows the possibilities for continuous improvement of professional, personal, and social qualifications.
- Understands the need and knows the possibilities for continuously enhancing their own competences.
- Is aware of the importance and understands non-technical aspects and consequences of the work of a computer engineer.

Program Content

1. Definition, elements, and examples of programming languages. Language reports. High-level programming languages. Programming techniques – structured vs. object-oriented.
2. General structure of a computer program. Source and executable code. Translation methods. Compilation vs. interpretation of code.
3. Program construction. Data types. Constants, variables, arrays. Arithmetic and logical operators. Assignment instructions, conditional instructions: single and multiple selection.
4. Organizing repetition operations – iteration. Loop iteration counters. Increment vs. decrement. Variations of iterative instructions.

Teaching Methods:

- Lectures
- Project methods
- Auditory / laboratory exercises

Education verification method: exam, activity during classes

Basics of programming 1 – project

The aim of the course is to gain knowledge in the field of the subject. The students will learn to design, design computer programs and create the correct source code, as well as run the created program code using the selected translator.

Learning Outcomes

in the terms of knowledge

- Achieves knowledge in the subject area, including building and developing computer programs.
- Knows and understands methods for analyzing the given problem.
- Knows and understands complex relationships between analyzed data.

in the terms of skills

- Achieves the ability to design correct source code for computer programs. Possesses skills in using selected compilers and tools for program development.

- Can design and program uncomplicated IT issues.
- Can use available literature to formulate and solve IT problems.

in the terms of social competencies

- Understands the need and knows the possibilities for continuous improvement of professional, personal, and social qualifications.
- Understands the need and knows the possibilities for continuously enhancing their own competences.
- Is aware of the importance and understands non-technical aspects and consequences of the work of a computer engineer.

Program Content

1. Definition, elements, and examples of programming languages. Language reports. High-level programming languages. Programming techniques – structured vs. object-oriented.
2. General structure of a computer program. Source and executable code. Translation methods. Compilation vs. interpretation of code.
3. Program construction. Data types. Constants, variables, arrays. Arithmetic and logical operators. Assignment instructions, conditional instructions: single and multiple selection.
4. Organizing repetition operations – iteration. Loop iteration counters. Increment vs. decrement. Variations of iterative instructions.

Teaching Methods:

- Lectures
- Project methods
- Auditory / laboratory exercises

Education verification method: exam, activity during classes

ZDW: Fundamentals of Philosophy / History of Philosophy- lecture

The Aims of the Subject

- Introduce students to the key philosophical concepts and issues in the history of philosophy.
- Equip students with knowledge about the main positions, concepts, and representatives of various philosophical schools throughout history.
- Develop a reflective and critical attitude towards different intellectual, ethical, and religious traditions.
- Enhance the ability to engage in discussion and present one's own views on a given topic in relation to selected philosophical concepts.

Learning Outcomes

in the terms of knowledge

- Defines selected philosophical concepts and explains the philosophical sources of educational terminology.
- Recognizes the connection between pedagogical doctrines and various philosophical perspectives and schools.
- Defines, distinguishes, and describes selected philosophical concepts of humanity.

in the terms of skills

- Can use philosophical knowledge to interpret pedagogical, social, educational, and cultural situations.
- Can apply knowledge of key philosophical problems and schools to analyze motives and patterns of human behavior.

in the terms of social competencies

- Recognizes the ethical dilemmas present in the field of scientific research.

Program Content

1. Philosophy and its branches (ontology, axiology, ethics, epistemology, logic); philosophical reflection. Borderline situations. Sources of philosophy – wonder, suffering.
2. Between myth and logos. Presocratic philosophy (philosophy of nature, relativism, variability, determinism, dialectical method of understanding reality, paradoxical issues).
3. Socrates (ethical intellectualism, dialogue) and sophists (relativism, utilitarianism, eristics, rhetoric, democracy). Plato – objective idealism, dualism, ethical absolutism, Plato's social philosophy. Aristotle – moderate realism, the ethics of the golden mean.
4. Hellenistic-Roman philosophy: cynics, cyrenaics, stoics, epicureans.
5. Medieval philosophy – theocentric era.
6. Empiricism (genetic and methodological), rationalism (nativism and apriorism), transcendental aesthetics of Kant, contemporary philosophy of science.
7. Ethics: absolutism, consequentialism, utilitarianism, relativism; ethical dilemmas of contemporary times.
8. Aesthetics as a sub-discipline of axiology, theory of art.
9. Problems of contemporary philosophy – bioethics, terrorism, ecology, feminism.

Teaching Methods:

- Conversational lecture
- Didactic discussion
- Analysis and interpretation of source texts
- Project method

Education verification method: exam, activity during classes

ZDW: Fundamentals of Philosophy / History of Philosophy- workshop

The Aims of the Subject

- Introduce students to the key philosophical concepts and issues in the history of philosophy.
- Equip students with knowledge about the main positions, concepts, and representatives of various philosophical schools throughout history.
- Develop a reflective and critical attitude towards different intellectual, ethical, and religious traditions.
- Enhance the ability to engage in discussion and present one's own views on a given topic in relation to selected philosophical concepts.

Learning Outcomes

in the terms of knowledge

- Defines selected philosophical concepts and explains the philosophical sources of educational terminology.
- Recognizes the connection between pedagogical doctrines and various philosophical perspectives and schools.
- Defines, distinguishes, and describes selected philosophical concepts of humanity.

in the terms of skills

- Can use philosophical knowledge to interpret pedagogical, social, educational, and cultural situations.
- Can apply knowledge of key philosophical problems and schools to analyze motives and patterns of human behavior.

in the terms of social competencies

- Recognizes the ethical dilemmas present in the field of scientific research.

Program Content

1. Philosophy and its branches (ontology, axiology, ethics, epistemology, logic); philosophical reflection. Borderline situations. Sources of philosophy – wonder, suffering.

2. Between myth and logos. Presocratic philosophy (philosophy of nature, relativism, variability, determinism, dialectical method of understanding reality, paradoxical issues).

- Socrates (ethical intellectualism, dialogue) and sophists (relativism, utilitarianism, eristics, rhetoric, democracy). Plato – objective idealism, dualism, ethical absolutism, Plato's social philosophy. Aristotle – moderate realism, the ethics of the golden mean.

- Hellenistic-Roman philosophy: cynics, cyrenaics, stoics, epicureans.
- Medieval philosophy – theocentric era.
- Empiricism (genetic and methodological), rationalism (nativism and apriorism), transcendental aesthetics of Kant, contemporary philosophy of science.
- Ethics: absolutism, consequentialism, utilitarianism, relativism; ethical dilemmas of contemporary times.
- Aesthetics as a sub-discipline of axiology, theory of art.
- Problems of contemporary philosophy – bioethics, terrorism, ecology, feminism.

Teaching Methods:

- Didactic discussion
- Analysis and interpretation of source texts
- Conversational lecture
- Project method

Education verification method: exam, activity during classes

Personal skills: Communication

The Aims of the Subject

- Introduce students to selected tools for effective communication (paraphrasing, "I" statements, naming feelings, naming emotions, open/closed questions, feedback).
- Improve skills in selecting an appropriate communication style depending on the situation.
- Develop the ability to communicate effectively in the process of solving individual and group problems.

Learning Outcomes

in the terms of skills

- Chooses an effective communication strategy for oneself.
- Applies selected tools of effective communication.
- Can appropriately present oneself in different situations.
- Organizes team work.

in the terms of social competencies

- Shows openness to resolving individual and group communication problems.
- Engages in teamwork by performing various group roles.

Program Content

1. Methods for effective contact with another person.
2. Tools for effective communication: naming feelings, using open questions, active listening: paraphrasing, reflecting, message precision, "I" statements.

3. The role of verbal and non-verbal communication.
4. Communication barriers.
5. The role and significance of emotions in the process of communication and integration.
6. The importance of integration and effective communication in the functioning of a group.

Teaching Methods:

- Didactic discussion
- Brainstorming
- Drama techniques
- Workshop method

Education verification method: exam, activity during classes

Information technology

The Aims of the Subject

- Introduce students to selected concepts related to information technology, multimedia, and media convergence.
- Equip students with skills to use tools and applications necessary for office work.
- Develop skills in interpreting and creating multimedia messages.

Learning Outcomes

in the terms of knowledge

- Describes and explains terminology related to information technology, multimedia, and media convergence, characterizes their sources and applications in management. Understands methods and principles of creating multimedia presentations, text documents, and spreadsheet management.
- Identifies aspects of intellectual property protection and copyright in the context of administrative work. Has knowledge of correct information retrieval on the Internet and using online services.

1

in the terms of skills

- Utilizes IT resources for self-development. Applies correct methods for information retrieval on the Internet and using online services.
- Uses specialized language and communicates precisely and coherently using various channels and communication techniques.
- Selects, constructs, and utilizes available materials, tools, and methods using modern technologies, with an understanding of intellectual property protection laws. Can create a correct multimedia presentation, text document, and spreadsheet.

in the terms of social competences

- Is aware of their level of knowledge and skills concerning the application of modern technologies in management, understands the need for continuous professional development and personal growth, especially in deepening knowledge and skills using modern technologies.
- Identifies and formulates moral and ethical issues related to their own and others' work, and recognizes violations of ethical standards in their work concerning intellectual property protection.

Program Content

1. Concepts related to information technology, multimedia, and media convergence. The role of information technology in managerial work.
2. Using IT tools and applications in education. Editing text documents: entering, correcting, proofreading, autocorrect, formatting, embedding objects in text, lists, tables, headers, sections, page numbering, print preview. Mail merge. Working with multi-page documents: footnotes, bookmarks, hyperlinks, table of contents, bibliography, indexes, etc.
3. Using IT tools and applications in education. Creating spreadsheets: data types, operators, arithmetic expressions, logical and text expressions, function arguments, function values, extracting parameters in task solutions, addressing methods, formulas, built-in functions, auto-fill, cell and range formatting, XY charts. Spreadsheet as a simple database - forms, searching, filtering, multi-field sorting.
4. Using IT tools and applications in education – creating multimedia presentations (PowerPoint and Prezi). Presentation design principles. Graphics, sound, animation of elements, adding hyperlinks, charts, presentation templates, presentation organization, automatic presentation. Saving presentations in different formats.
5. Interpreting and creating selected educational multimedia messages.
6. Intellectual property protection in IT in education: creating, sharing, and using source materials, types of licenses.

Teaching Methods:

- Demonstrations with explanations
- Instruction
- Practical exercises
- Individual work

Education verification method: exam, activity during classes

Architecture of computer systems / Functionality, organization and implementation of computer systems – lecture

The aim of the course is to gain knowledge about the construction of computers and methods of data presentation and processing in computer, as well as low-level mechanisms of operation, construction and creation of computer programs.

The aim of the subject is:

Familiarize students with the construction and basic components of computers.

Teach students how data is represented and processed within a computer system.

Introduce students to low-level mechanisms of computer operation, including the construction and creation of programs.

Learning Outcomes

In the terms of knowledge

- Gain knowledge of the history of computing devices and understand computer architecture and construction.
- Understand key concepts related to programming in assembly language (x86 processors).
- Understand how computer programs operate.

in the terms of skills

- Ability to design applications in assembly language, including sections and data declarations.
- Ability to apply knowledge to solve complex and atypical computer science problems.
- Ability to use available literature to formulate and solve computer science problems.

in the terms of social competencies

- Understand the need and opportunities for continuous professional, personal, and social development.
- : Recognize the need and opportunities for ongoing improvement of personal competencies.
- Awareness of the importance and understanding of non-technical aspects and consequences of the work of a computer engineer.

Course Content

1. **History of Counting and Computing Devices**:
   - Introduction to the development and history of computational technology.
2. **Classical Von Neumann Computer Architecture**:
   - Understanding the architecture of computers according to the Von Neumann model.
3. **Number Systems**:
   - Learning about the number systems used in computers.
4. **Information Coding**:
   - Understanding methods of information coding in computer systems.

5. **Basics of Logic Circuits**:
    - Learning about the fundamentals of logic circuits used in computers.
6. **Arithmetic Foundations of Computer Operation**:
    - Understanding the basics of computer arithmetic.
7. **Concept of Microprocessor Systems**:
    - Introduction to microprocessor systems and their operation.
8. **Operational Memory**:
    - Understanding the function and structure of operational memory in computers.
9. **External Devices**:
    - Introduction to various peripheral devices for computers.
10. **Processors**:
    - Understanding the operation and architecture of computer processors.
11. **Elements of Computer Software**:
    - Learning about components of computer software.
12. **Computer Classification**:
    - Understanding different types and classifications of computers.
13. **Pipelined Architectures**:
    - Learning about pipelined architectures in computers.
14. **Parallel Computers and Supercomputers**:
    - Understanding the operation of parallel computers and supercomputers.
15. **Theoretical Foundations of Assembly Language Programming**:
    - Introduction to basic principles of programming in assembly language, including sections and data declarations.

Teaching Methods

- **Lecture**: Traditional method of delivering theoretical content.
- **Multimedia Techniques**: Use of various media formats (e.g., video, animations) to enhance understanding and learning.

Education verification method: exam, activity during classes

Architecture of computer systems / Functionality, organization and implementation of computer systems – workshops

The aim of the course is to gain knowledge about the construction of computers and methods of data presentation and processing in computer, as well as low-level mechanisms of operation, construction and creation of computer programs.

The aim of the subject is:

Familiarize students with the construction and basic components of computers.

Teach students how data is represented and processed within a computer system.

Introduce students to low-level mechanisms of computer operation, including the construction and creation of programs.

Learning Outcomes

In the terms of knowledge

- Gain knowledge of the history of computing devices and understand computer architecture and construction.
- Understand key concepts related to programming in assembly language (x86 processors).
- Understand how computer programs operate.

in the terms of skills

- Ability to design applications in assembly language, including sections and data declarations.
- Ability to apply knowledge to solve complex and atypical computer science problems.
- Ability to use available literature to formulate and solve computer science problems.

in the terms of social competencies

- Understand the need and opportunities for continuous professional, personal, and social development.
- : Recognize the need and opportunities for ongoing improvement of personal competencies.
- Awareness of the importance and understanding of non-technical aspects and consequences of the work of a computer engineer.

Course Content

1. **History of Counting and Computing Devices**:
2. Introduction to the development and history of computational technology.

- **Classical Von Neumann Computer Architecture**:
    1. Understanding the architecture of computers according to the Von Neumann model.
- **Number Systems**:
    1. Learning about the number systems used in computers.
- **Information Coding**:
    1. Understanding methods of information coding in computer systems.
- **Basics of Logic Circuits**:
    1. Learning about the fundamentals of logic circuits used in computers.
- **Arithmetic Foundations of Computer Operation**:
    1. Understanding the basics of computer arithmetic.
- **Concept of Microprocessor Systems**:
    1. Introduction to microprocessor systems and their operation.
- **Operational Memory**:
    1. Understanding the function and structure of operational memory in computers.
- **External Devices**:
    1. Introduction to various peripheral devices for computers.

- **Processors**:
    1. Understanding the operation and architecture of computer processors.
- **Elements of Computer Software**:
    1. Learning about components of computer software.
- **Computer Classification**:
    1. Understanding different types and classifications of computers.
- **Pipelined Architectures**:
    1. Learning about pipelined architectures in computers.
- **Parallel Computers and Supercomputers**:
    1. Understanding the operation of parallel computers and supercomputers.
- **Theoretical Foundations of Assembly Language Programming**:
    1. Introduction to basic principles of programming in assembly language, including sections and data declarations.

Teaching Methods

Lecture: Traditional method of delivering theoretical content.

Multimedia Techniques: Use of various media formats (e.g., video, animations) to enhance understanding and learning

Education verification method: exam, activity during classes

2nd year, 3th semester

ZDW: SQL Language / Database Management System – lecture

The aim of the course is to gain knowledge in the field of the study. The students will learn to process data in a relational database using SQL.

Aims of the Subject: The aim is to equip students with the ability to define tasks within a database management system.

Learning Outcomes:

In terms of Knowledge: Students will be able to:

- Identify areas of application for databases.
- Recognize tools dedicated to designing, collecting, and exploring data.
- Define the benefits of using databases in engineering and business practices.
- Formulate data models that describe objects and processes occurring in practice.
- Apply appropriate techniques for implementing data models.

In terms of Skills: Students will be able to:

- Design database structures and schemas.
- Implement and design databases.
- Create data integrity mechanisms.

- Develop queries for databases.
- Manipulate data effectively.

In terms of Social Competencies: Students will:

- Approach data model building, database design, implementation, and exploration with creativity.
- Actively use databases in engineering and business practices.
- Be oriented towards the effective use of database design and exploration tools.
- Be aware of the limitations of data models and their implementation.
-

Program Content:

- Structure of database management systems (DBMS) - characteristics of individual modules, types of inputs.
- Basic functions of DBMS and their implementations. Features of DBMS compared to file management systems.
- Transaction management - properties of transactions (ACID principles), transaction operations and history, concurrent transactions, and their levels.
- The role of the scheduler and data manager. Two-phase locking protocol, transaction deadlock, and locking with various isolation levels.
- The timestamp method. Multiversion algorithms - synchronization mechanisms and popular variants of the method (two-phase locking multiversion and timestamp multiversion algorithms).
- Integrity vs. database consistency. Levels and protection of integrity.
- Types of integrity constraints and their verification.
- Query processing and evaluation - query decomposition, rules for transforming relational algebra operations.
- Estimating the cost of relational algebra operations, database statistics. Query optimization - heuristic methods.

Teaching Methods:

- Lectures
- Project-based methods
- Auditorium/laboratory exercises

Education verification method: exam, activity during classes

ZDW: SQL Language / Database Management System – tutorials

Aims of the Subject: The aim is to equip students with the ability to define tasks within a database management system.

Learning Outcomes:

In terms of Knowledge: Students will be able to:

- Identify areas of application for databases.

- Recognize tools dedicated to designing, collecting, and exploring data.
- Define the benefits of using databases in engineering and business practices.
- Formulate data models that describe objects and processes occurring in practice.
- Apply appropriate techniques for implementing data models.

In terms of Skills: Students will be able to:

- Design database structures and schemas.
- Implement and design databases.
- Create data integrity mechanisms.
- Develop queries for databases.
- Manipulate data effectively.

In terms of Social Competencies: Students will:

- Approach data model building, database design, implementation, and exploration with creativity.
- Actively use databases in engineering and business practices.
- Be oriented towards the effective use of database design and exploration tools.
- Be aware of the limitations of data models and their implementation.

Program Content:

- Structure of database management systems (DBMS) - characteristics of individual modules, types of inputs.
- Basic functions of DBMS and their implementations. Features of DBMS compared to file management systems.
- Transaction management - properties of transactions (ACID principles), transaction operations and history, concurrent transactions, and their levels.
- The role of the scheduler and data manager. Two-phase locking protocol, transaction deadlock, and locking with various isolation levels.
- The timestamp method. Multiversion algorithms - synchronization mechanisms and popular variants of the method (two-phase locking multiversion and timestamp multiversion algorithms).
- Integrity vs. database consistency. Levels and protection of integrity.
- Types of integrity constraints and their verification.
- Query processing and evaluation - query decomposition, rules for transforming relational algebra operations.
- Estimating the cost of relational algebra operations, database statistics. Query optimization - heuristic methods.

Teaching Methods:

- Lectures
- Project-based methods
- Auditorium/laboratory exercises

Education verification method: exam, activity during classes

ZDW: SQL Language / Database Management System – project

Aims of the Subject: The aim is to equip students with the ability to define tasks within a database management system.

Learning Outcomes:

In terms of Knowledge: Students will be able to:

- Identify areas of application for databases.
- Recognize tools dedicated to designing, collecting, and exploring data.
- Define the benefits of using databases in engineering and business practices.
- Formulate data models that describe objects and processes occurring in practice.
- Apply appropriate techniques for implementing data models.

In terms of Skills: Students will be able to:

- Design database structures and schemas.
- Implement and design databases.
- Create data integrity mechanisms.
- Develop queries for databases.
- Manipulate data effectively.

In terms of Social Competencies: Students will:

- Approach data model building, database design, implementation, and exploration with creativity.
- Actively use databases in engineering and business practices.
- Be oriented towards the effective use of database design and exploration tools.
- Be aware of the limitations of data models and their implementation.

Program Content:

- Structure of database management systems (DBMS) - characteristics of individual modules, types of inputs.
- Basic functions of DBMS and their implementations. Features of DBMS compared to file management systems.
- Transaction management - properties of transactions (ACID principles), transaction operations and history, concurrent transactions, and their levels.
- The role of the scheduler and data manager. Two-phase locking protocol, transaction deadlock, and locking with various isolation levels.
- The timestamp method. Multiversion algorithms - synchronization mechanisms and popular variants of the method (two-phase locking multiversion and timestamp multiversion algorithms).
- Integrity vs. database consistency. Levels and protection of integrity.
- Types of integrity constraints and their verification.
- Query processing and evaluation - query decomposition, rules for

transforming relational algebra operations.

- Estimating the cost of relational algebra operations, database statistics. Query optimization - heuristic methods.

Teaching Methods:

- Lectures
- Project-based methods
- Auditorium/laboratory exercises

Education verification method: exam, activity during classes

Database systems – lectures

The aim of the course is to gain knowledge in the field of advanced software functionalities for data collection and mining. The course will prepare students for the use of IT tools supporting the collection and data mining, and show how to design databases and explore them. The students will develop the ability to use IT tools dedicated to data collection and mining, as well as shape the attitude of professional use of databases in engineering and business practice.

Aims of the Subject:

- Acquire skills in utilizing advanced functionalities of data collection and exploration programs.
- Prepare students for the use of IT tools supporting data collection and exploration.
- Equip students with the ability to design and explore databases.
- Develop skills in handling IT tools dedicated to data collection and exploration.
- Foster a professional attitude towards the use of databases in engineering and business practices.

Learning Outcomes:

In terms of Knowledge: Students will be able to:

- Identify areas of application for databases.
- Recognize tools dedicated to designing, collecting, and exploring data.
- Define the benefits of using databases in engineering and business practices.
- Formulate data models that describe objects and processes occurring in practice.
- Apply appropriate techniques for implementing data models.

In terms of Skills: Students will be able to:

- Design database structures and schemas.
- Implement and design databases.

- Create mechanisms for data integrity.
- Develop queries for databases.
- Manipulate data effectively.

In terms of Social Competencies: Students will:

- Approach data model building, database design, implementation, and exploration with creativity.
- Actively use databases in engineering and business practices.
- Be oriented towards the effective use of database design and exploration tools.
- Be aware of the limitations of data models and their implementation.

Program Content:

- Practical overview of database systems – introduction.
- Data models.
- Database languages.
- Relational databases (RBD).
- Creating data structures in RBD.
- Data processing in RBD.
- Data exploration in RBD.
- Data manipulation in relational databases.
- Identifying the functionalities of the database management system used in laboratory sessions.
- Designing database models.
- Designing and creating database structures.
- Data integrity mechanisms.
- Importing data from external sources.
- Simple queries. Data sorting.
- Conditional data selection. Comparison operators.
- Table joins.
- Data grouping.
- Simple and correlated queries.
- Query optimization.
- Data manipulation – insertion, modification, deletion.

Teaching Methods:

- Laboratory exercises
- Workshop methods
- Didactic discussions
- Demonstrations

Education verification method: exam, activity during classes

Database systems – project

The aim of the course is to gain knowledge in the field of advanced software functionalities for data collection and mining. The course will prepare students for the use of IT tools supporting the collection and data mining, and show how to design databases and explore them. The students will develop the ability to use IT tools dedicated to data collection and mining, as well as shape the attitude of professional use of databases in engineering and business practice.

Aims of the Subject:

- Acquire skills in utilizing advanced functionalities of data collection and exploration programs.
- Prepare students for the use of IT tools supporting data collection and exploration.
- Equip students with the ability to design and explore databases.
- Develop skills in handling IT tools dedicated to data collection and exploration.
- Foster a professional attitude towards the use of databases in engineering and business practices.

Learning Outcomes:

In terms of Knowledge: Students will be able to:

- Identify areas of application for databases.
- Recognize tools dedicated to designing, collecting, and exploring data.
- Define the benefits of using databases in engineering and business practices.
- Formulate data models that describe objects and processes occurring in practice.
- Apply appropriate techniques for implementing data models.

In terms of Skills: Students will be able to:

- Design database structures and schemas.
- Implement and design databases.
- Create mechanisms for data integrity.
- Develop queries for databases.
- Manipulate data effectively.

In terms of Social Competencies: Students will:

- Approach data model building, database design, implementation, and exploration with creativity.
- Actively use databases in engineering and business practices.
- Be oriented towards the effective use of database design and exploration tools.
- Be aware of the limitations of data models and their implementation.

Program Content:

- Practical overview of database systems – introduction.
- Data models.

- Database languages.
- Relational databases (RBD).
- Creating data structures in RBD.
- Data processing in RBD.
- Data exploration in RBD.
- Data manipulation in relational databases.
- Identifying the functionalities of the database management system used in laboratory sessions.
- Designing database models.
- Designing and creating database structures.
- Data integrity mechanisms.
- Importing data from external sources.
- Simple queries. Data sorting.
- Conditional data selection. Comparison operators.
- Table joins.
- Data grouping.
- Simple and correlated queries.
- Query optimization.
- Data manipulation – insertion, modification, deletion.

Teaching Methods:

- Laboratory exercises
- Workshop methods
- Didactic discussions
- Demonstrations

Education verification method: exam, activity during classes

Database systems – workshops

The aim of the course is to gain knowledge in the field of advanced software functionalities for data collection and mining. The course will prepare students for the use of IT tools supporting the collection and data mining, and show how to design databases and explore them. The students will develop the ability to use IT tools dedicated to data collection and mining, as well as shape the attitude of professional use of databases in engineering and business practice.

Aims of the Subject:

- Acquire skills in utilizing advanced functionalities of data collection and exploration programs.
- Prepare students for the use of IT tools supporting data collection and exploration.
- Equip students with the ability to design and explore databases.
- Develop skills in handling IT tools dedicated to data collection and exploration.
- Foster a professional attitude towards the use of databases in engineering and business practices.

Learning Outcomes:

In terms of Knowledge: Students will be able to:

- Identify areas of application for databases.
- Recognize tools dedicated to designing, collecting, and exploring data.
- Define the benefits of using databases in engineering and business practices.
- Formulate data models that describe objects and processes occurring in practice.
- Apply appropriate techniques for implementing data models.

In terms of Skills: Students will be able to:

- Design database structures and schemas.
- Implement and design databases.
- Create mechanisms for data integrity.
- Develop queries for databases.
- Manipulate data effectively.

In terms of Social Competencies: Students will:

- Approach data model building, database design, implementation, and exploration with creativity.
- Actively use databases in engineering and business practices.
- Be oriented towards the effective use of database design and exploration tools.
- Be aware of the limitations of data models and their implementation.

Program Content:

- Practical overview of database systems – introduction.
- Data models.
- Database languages.
- Relational databases (RBD).
- Creating data structures in RBD.
- Data processing in RBD.
- Data exploration in RBD.
- Data manipulation in relational databases.
- Identifying the functionalities of the database management system used in laboratory sessions.
- Designing database models.
- Designing and creating database structures.
- Data integrity mechanisms.
- Importing data from external sources.
- Simple queries. Data sorting.
- Conditional data selection. Comparison operators.
- Table joins.
- Data grouping.
- Simple and correlated queries.
- Query optimization.

- Data manipulation – insertion, modification, deletion.

Teaching Methods:

- Laboratory exercises
- Workshop methods
- Didactic discussions
- Demonstrations

Education verification method: exam, activity during classes

Basics of computer graphics – lecture

The aim of the course is to familiarize students with the basic elements of computer graphics and their practical applications.

Aims of the Subject:

- To familiarize students with the fundamental elements of computer graphics and their practical applications.

Learning Outcomes:

In terms of Knowledge: Students will:

- Have a basic understanding of the theory related to computer graphics.
- Know programming languages and understand which language to use for solving specific problems.
- Be knowledgeable about the capabilities of programming tools that assist in the creation of computer graphics.

In terms of Skills: Students will be able to:

- Gain practical skills in programming 2D raster and vector graphics.
- Gain practical skills in programming 3D raster and vector graphics.
- Acquire practical skills in selecting graphical tools and libraries that support graphic programming.

In terms of Social Competencies: Students will:

- Demonstrate the ability for continuous learning, improvement, and enhancement of professional, personal, and social competencies.

Program Content:

- Basics of programming raster and vector graphics, both 2D and 3D.
- Color and color theory in computer graphics.
- Basic issues of geometric transformations.

- Methods of processing and editing raster graphics.
- Tools and methods for programming graphic editing.
- Analysis of tools and libraries supporting graphic programming.

Teaching Methods:

- Lecture
- Conversational lecture
- Brainstorming
- Project work

Education verification method: exam, activity during classes

Basics of computer graphics – workshops

The aim of the course is to familiarize students with the basic elements of computer graphics and their practical applications.

Aims of the Subject:

- To familiarize students with the fundamental elements of computer graphics and their practical applications.

Learning Outcomes:

In terms of Knowledge: Students will:

- Have a basic understanding of the theory related to computer graphics.
- Know programming languages and understand which language to use for solving specific problems.
- Be knowledgeable about the capabilities of programming tools that assist in the creation of computer graphics. **2**

In terms of Skills: Students will be able to:

- Gain practical skills in programming 2D raster and vector graphics.
- Gain practical skills in programming 3D raster and vector graphics.
- Acquire practical skills in selecting graphical tools and libraries that support graphic programming.

In terms of Social Competencies: Students will:

- Demonstrate the ability for continuous learning, improvement, and enhancement of professional, personal, and social competencies.

Program Content:

- Basics of programming raster and vector graphics, both 2D and 3D.

- Color and color theory in computer graphics.
- Basic issues of geometric transformations.
- Methods of processing and editing raster graphics.
- Tools and methods for programming graphic editing.
- Analysis of tools and libraries supporting graphic programming.

Teaching Methods:

- Lecture
- Conversational lecture
- Brainstorming
- Project work

Education verification method: exam, activity during classes

Basics of computer graphics -project

The aim of the course is to familiarize students with the basic elements of computer graphics and their practical applications.

The aim of the course is to familiarize students with the basic elements of computer graphics and their practical applications.

Aims of the Subject:

- To familiarize students with the fundamental elements of computer graphics and their practical applications.

Learning Outcomes:

In terms of Knowledge: Students will:

3

- Have a basic understanding of the theory related to computer graphics.
- Know programming languages and understand which language to use for solving specific problems.
- Be knowledgeable about the capabilities of programming tools that assist in the creation of computer graphics.

In terms of Skills: Students will be able to:

- Gain practical skills in programming 2D raster and vector graphics.
- Gain practical skills in programming 3D raster and vector graphics.
- Acquire practical skills in selecting graphical tools and libraries that support graphic programming.

In terms of Social Competencies: Students will:

- Demonstrate the ability for continuous learning, improvement, and enhancement of professional, personal, and social competencies.

Program Content:

- Basics of programming raster and vector graphics, both 2D and 3D.
- Color and color theory in computer graphics.
- Basic issues of geometric transformations.
- Methods of processing and editing raster graphics.
- Tools and methods for programming graphic editing.
- Analysis of tools and libraries supporting graphic programming.

Teaching Methods:

- Lecture
- Conversational lecture
- Brainstorming
- Project work

Education verification method: exam, activity during classes

Management systems

Aims of the Subject:

- To provide knowledge in the subject area.
- To offer insights into business management.
- To familiarize students with major management systems.

Learning Outcomes:

In terms of Knowledge: Students will:

- Have a basic understanding of the theory related to the subject.
- Know and understand methods of analysis related to the subject.
- Understand selected issues in the field of knowledge, including methods of analysis. 1

In terms of Skills: Students will be able to:

- Apply acquired knowledge to solve complex IT problems.
- Select appropriate tests to verify the correctness of a project.
- Utilize available literature sources to formulate and solve IT problems.

In terms of Social Competencies: Students will:

- Apply acquired knowledge to solve complex IT problems.
- Choose appropriate tests to ensure project accuracy.
- Use available literature to formulate and resolve IT issues.

Program Content:

- Characteristics of information systems in enterprises.
- The essence of ERP systems, including system modules (personnel, payroll, sales, etc.), examples of implementations, distribution models, and data security.
- Fundamentals of designing system infrastructures and management processes.
- Process-based view of organizations and their reflection in ERP system modules.

Teaching Methods:

- Lectures
- Project-based methods
- Auditory/laboratory exercises

Education verification method: exam, activity during classes

Business process modeling- project

Aims of the Subject:

- To provide knowledge in the subject area.
- To introduce the concept of business simulation modeling.
- To prepare students to use computer simulation methods.

Learning Outcomes:

In terms of Knowledge: Students will:

- Have a basic understanding of the theory related to the subject.
- Know and understand methods of analysis related to the subject.
- Understand selected issues in the field of knowledge, including methods of analysis.

In terms of Skills: Students will be able to:

- Apply acquired knowledge to solve complex IT problems.
- Select appropriate tests to verify the correctness of a project.
- Utilize available literature sources to formulate and solve IT problems.

In terms of Social Competencies: Students will:

- Apply acquired knowledge to solve complex IT problems.
- Choose appropriate tests to ensure project accuracy.
- Use available literature to formulate and resolve IT issues.

Program Content:

- Process-oriented approach in modern organizations.
- Basics of business process management in organizations.
- Methodology of business process modeling.
- Simulation of business processes.

Teaching Methods:

- Lectures
- Project-based methods
- Auditory/laboratory exercises

Education verification method: exam, activity during classes

Business process modeling- workshops

Aims of the Subject:

- To provide knowledge in the subject area.
- To introduce the concept of business simulation modeling.
- To prepare students to use computer simulation methods.

Learning Outcomes:

In terms of Knowledge: Students will:

- Have a basic understanding of the theory related to the subject.
- Know and understand methods of analysis related to the subject.
- Understand selected issues in the field of knowledge, including methods of analysis.

In terms of Skills: Students will be able to:

- Apply acquired knowledge to solve complex IT problems.
- Select appropriate tests to verify the correctness of a project.
- Utilize available literature sources to formulate and solve IT problems.

In terms of Social Competencies: Students will:

- Apply acquired knowledge to solve complex IT problems.
- Choose appropriate tests to ensure project accuracy.
- Use available literature to formulate and resolve IT issues.

Program Content:

- Process-oriented approach in modern organizations.
- Basics of business process management in organizations.
- Methodology of business process modeling.
- Simulation of business processes.

Teaching Methods:

- Lectures
- Project-based methods
- Auditory/laboratory exercises

Education verification method: exam, activity during classes

Basics of creativity 2

Course Objectives

- Introduce students to concepts related to creativity.
- Equip students with knowledge about the language of intentional creativity.
- Familiarize students with the subjective concept of a human being.
- Develop fluency, flexibility, and originality in thinking.
- Enhance the ability to perceive and assign new meanings and interpretations to reality.
- Prepare students for creative problem-solving.

Learning Outcomes

In terms of knowledge:

- Students can list differences between heteronomous/reductionist views and a subjective view of a human being.
- Students can differentiate between creative and standard behaviors.
- Students can explain concepts related to the creative development of individuals.
- Students can define creative, intentionally creative, and standard actions.
- Students understand the evolution of the meaning and applicability of the concept of creativity.
- Students can identify what fluency, flexibility, and originality in thinking are.
- Students can explain selected methods of creative problem-solving.
- Students can describe the process of problem-solving according to the techne method.
- Students can present the advantages and disadvantages of the techne method in problem-solving.

In terms of skills:

- Students are able to modify their perception of reality.
- Students can combine different ideas, concepts, and solutions.
- Students can justify their opinions and viewpoints.
- Students demonstrate readiness to break free from conventional thinking and action patterns.
- Students can apply selected methods of creative problem-solving.

In terms of social competencies:

- Students take care of their autonomy in thinking and acting.

- Students organize their activities in an innovative way.
- Students demonstrate flexibility in thinking and acting.

Course Content

1. The history of the concept of creativity.
2. Understanding the concept of creativity – various theories and approaches.
3. The techne method and scientism.
4. Heuristics – thinking through analogy, metaphor, and abstraction.
5. Selected methods of creative problem-solving.
6. The subjective vision of a human being.
7. Conscious development of creative dispositions.

Teaching Methods

- Didactic discussion
- Brainstorming
- Workshop method

Education verification method: exam, activity during classes

3rd year, 5th semester

ZDW: Programming of web applications / Framework for web applications – lecture

The aim of the study transferring knowledge in the field of the subject. The students will learn the ability to program and design web applications - run on web pages /The aim of the course is to gain knowledge about the use of Frameworks. The students will be acquainted with the knowledge of the .NET Framework and other technologies, how to create applications using Ruby on Rails Framework, creating scripts, functional programming, creating systems in the cloud.

Aims of the Subject

- Providing knowledge in the subject area.
- Achieving the ability to program and design web applications that run on web pages.

Learning Outcomes

- **Knowledge**
  - Acquires knowledge in the subject area, including the ability to design and program web applications.
  - Knows programming languages and understands which language to use for solving selected IT problems.
  - Knows and understands selected issues in the field, including methods of modeling.

- **Skills**
  - Can apply appropriate programming languages and development tools for building web applications and web services.
  - Can use knowledge to solve complex and unconventional IT problems.
  - Can utilize available literature to formulate and solve IT problems.
- **Social Competencies**
  - Understands the need for and knows the possibilities of continuous professional, personal, and social development.
  - Understands the need for and knows the possibilities of continuous self-improvement.
  - Is aware of the importance and understands the non-technical aspects and consequences of the work of an IT engineer.

Program Content

1. Overview and development of internet programming technologies: HTML, XHTML, Java, PHP, etc.
2. Development of HTML as a tool for building web applications. Characteristics of PHP technology – its origins, development, language elements, and applications.
3. Characteristics of the Common Gateway Interface (CGI) model.
4. Servlet and applet applications. The three-tier model of web application operation in software engineering. Data access layer in the model.
5. The significance of object-oriented (OOA/D) approach to building web applications. COM/CORBA models – Component Object Model/Common Object Request Broker Architecture.
6. Database management technologies in the web. Development of SQL as a tool for developing web applications for database management.

Teaching Methods

- Lectures
- Project-based methods
- Laboratory exercises
- Seminars / didactic discussions

Education verification method: exam, activity during classes

ZDW: Programming of web applications / Framework for web applications – workshops

The aim of the study transferring knowledge in the field of the subject. The students will learn the ability to program and design web applications - run on web pages /The aim of the course is to gain knowledge about the use of Frameworks. The students will be acquainted with the knowledge of the .NET Framework and other technologies, how to create applications using Ruby on Rails Framework, creating scripts, functional programming, creating systems in the cloud.

2

Aims of the Subject

- Providing knowledge in the subject area.
- Achieving the ability to program and design web applications that run on web pages.

Learning Outcomes

- **Knowledge**
  - Acquires knowledge in the subject area, including the ability to design and program web applications.
  - Knows programming languages and understands which language to use for solving selected IT problems.
  - Knows and understands selected issues in the field, including methods of modeling.
- **Skills**
  - Can apply appropriate programming languages and development tools for building web applications and web services.
  - Can use knowledge to solve complex and unconventional IT problems.
  - Can utilize available literature to formulate and solve IT problems.
- **Social Competencies**
  - Understands the need for and knows the possibilities of continuous professional, personal, and social development.
  - Understands the need for and knows the possibilities of continuous self-improvement.
  - Is aware of the importance and understands the non-technical aspects and consequences of the work of an IT engineer.

Program Content

1. Overview and development of internet programming technologies: HTML, XHTML, Java, PHP, etc.
2. Development of HTML as a tool for building web applications. Characteristics of PHP technology – its origins, development, language elements, and applications.
3. Characteristics of the Common Gateway Interface (CGI) model.
4. Servlet and applet applications. The three-tier model of web application operation in software engineering. Data access layer in the model.
5. The significance of object-oriented (OOA/D) approach to building web applications. COM/CORBA models – Component Object Model/Common Object Request Broker Architecture.
6. Database management technologies in the web. Development of SQL as a tool for developing web applications for database management.

Teaching Methods

- Lectures
- Project-based methods

- Laboratory exercises
- Seminars / didactic discussions

Education verification method: exam, activity during classes

ZDW: Programming of web applications / Framework for web applications – project

The aim of the study transferring knowledge in the field of the subject. The students will learn the ability to program and design web applications - run on web pages /The aim of the course is to gain knowledge about the use of Frameworks. The students will be acquainted with the knowledge of the .NET Framework and other technologies, how to create applications using Ruby on Rails Framework, creating scripts, functional programming, creating systems in the cloud.

Aims of the Subject

- Providing knowledge in the subject area.
- Achieving the ability to program and design web applications that run on web pages.

Learning Outcomes

Knowledge

- Acquires knowledge in the subject area, including the ability to design and program web applications. 2
- Knows programming languages and understands which language to use for solving selected IT problems.
- Knows and understands selected issues in the field, including methods of modeling.

Skills

- Can apply appropriate programming languages and development tools for building web applications and web services.
- Can use knowledge to solve complex and unconventional IT problems.
- Can utilize available literature to formulate and solve IT problems.

Social Competencies

- Understands the need for and knows the possibilities of continuous professional, personal, and social development.
- Understands the need for and knows the possibilities of continuous self-improvement.
- Is aware of the importance and understands the non-technical aspects

and consequences of the work of an IT engineer.

Program Content

1. Overview and development of internet programming technologies: HTML, XHTML, Java, PHP, etc.
2. Development of HTML as a tool for building web applications. Characteristics of PHP technology – its origins, development, language elements, and applications.
3. Characteristics of the Common Gateway Interface (CGI) model.
4. Servlet and applet applications. The three-tier model of web application operation in software engineering. Data access layer in the model.
5. The significance of object-oriented (OOA/D) approach to building web applications. COM/CORBA models – Component Object Model/Common Object Request Broker Architecture.
6. Database management technologies in the web. Development of SQL as a tool for developing web applications for database management.

Teaching Methods

- Lectures
- Project-based methods
- Laboratory exercises
- Seminars / didactic discussions

Education verification method: exam, activity during classes

Object oriented programming 2 – lecture

The aim of the course is to transfer knowledge in the field of the subject. The students will improve the ability to design computer programs in the OOP - Object Oriented Programming paradigm, creating effective source code and using the advantages of the object-oriented approach.

Aims of the Subject

- Ability to design a special-purpose information system – an intelligent information system. **2**
- Familiarize students with knowledge related to embedded systems and other technologies, script creation, functional programming, and cloud-based systems.

Learning Outcomes

Knowledge

- Understands information systems and the process of information management within an organization.

- Identifies tools dedicated to data design, collection, and exploration.
- Defines the benefits of using databases in engineering and business practice.
- Formulates data models that describe objects and processes occurring in practice.
- Applies appropriate techniques for implementing data models.

Skills

- Designs databases and their structures.
- Implements and designs databases.
- Creates data integrity mechanisms.
- Writes queries for databases.
- Manipulates data.

Social Competencies

- Approaches data model building, database design, implementation, and exploration creatively.
- Actively utilizes databases in engineering and business practice.
- Focuses on the effective use of database design and exploration tools.
- Is aware of the limitations of data models and their implementation.

Program Content

1. General characteristics of special-purpose systems. Characteristics of selected intelligent embedded systems. Real-life examples.
2. Designing an information system in conjunction with a specialized embedded component.
3. System installation and analysis of potential implementation errors.

Teaching Methods

- Laboratory exercises
- Workshop method
- Didactic discussions
- Project-based method
- Demonstrations

Education verification method: exam, activity during classes

Object oriented programming 2 – workshops

The aim of the course is to transfer knowledge in the field of the subject. The students will improve the ability to design computer programs in the OOP - Object Oriented Programming paradigm, creating effective source code and using the advantages of the object-oriented approach.

Aims of the Subject

- Ability to design a special-purpose information system – an intelligent information system.
- Familiarize students with knowledge related to embedded systems and other technologies, script creation, functional programming, and cloud-based systems.

Learning Outcomes

Knowledge

- Understands information systems and the process of information management within an organization.
- Identifies tools dedicated to data design, collection, and exploration.
- Defines the benefits of using databases in engineering and business practice.
- Formulates data models that describe objects and processes occurring in practice.
- Applies appropriate techniques for implementing data models.

Skills

- Designs databases and their structures.
- Implements and designs databases.
- Creates data integrity mechanisms.
- Writes queries for databases.
- Manipulates data.

Social Competencies

- Approaches data model building, database design, implementation, and exploration creatively.
- Actively utilizes databases in engineering and business practice.
- Focuses on the effective use of database design and exploration tools.
- Is aware of the limitations of data models and their implementation.

Program Content

1. General characteristics of special-purpose systems. Characteristics of selected intelligent embedded systems. Real-life examples.
2. Designing an information system in conjunction with a specialized embedded

component.
3. System installation and analysis of potential implementation errors.

Teaching Methods

- Laboratory exercises
- Workshop method
- Didactic discussions
- Project-based method
- Demonstrations

Education verification method: exam, activity during classes

Object oriented programming 2 – project

The aim of the course is to transfer knowledge in the field of the subject. The students will improve the ability to design computer programs in the OOP - Object Oriented Programming paradigm, creating effective source code and using the advantages of the object-oriented approach.

Aims of the Subject

- Ability to design a special-purpose information system – an intelligent information system.
- Familiarize students with knowledge related to embedded systems and other technologies, script creation, functional programming, and cloud-based systems.

Learning Outcomes

Knowledge

- Understands information systems and the process of information management within an organization.
- Identifies tools dedicated to data design, collection, and exploration.
- Defines the benefits of using databases in engineering and business practice.
- Formulates data models that describe objects and processes occurring in practice.
- Applies appropriate techniques for implementing data models.

Skills

- Designs databases and their structures.
- Implements and designs databases.
- Creates data integrity mechanisms.
- Writes queries for databases.
- Manipulates data.

Social Competencies

- Approaches data model building, database design, implementation, and exploration creatively.
- Actively utilizes databases in engineering and business practice.
- Focuses on the effective use of database design and exploration tools.
- Is aware of the limitations of data models and their implementation.

Program Content

1. General characteristics of special-purpose systems. Characteristics of selected intelligent embedded systems. Real-life examples.
2. Designing an information system in conjunction with a specialized embedded component.
3. System installation and analysis of potential implementation errors.

Teaching Methods

- Laboratory exercises
- Workshop method
- Didactic discussions
- Project-based method
- Demonstrations

Education verification method: exam, activity during classes

Online data base- lecture

Aims of the Subject

- Conveying knowledge in the subject area.
- Achieving the ability to design a practical relational database.
- Gaining proficiency in using a selected Database Management System (DBMS) for implementing a web-based practical database.

Learning Outcomes

Knowledge

- Gaining knowledge in database technology and database management systems on the global network.
- Understanding and knowing selected issues in data modeling.
- Knowing and understanding tools and data analysis languages (including SQL).

Skills

- Achieving the ability to design the structure of a practical relational database and selecting and using a database management system.
- Utilizing knowledge to solve complex and unusual IT problems.
- Using available literature sources for formulating and solving IT

problems.

Social Competencies

- Understanding the need and knowing the possibilities of continuously improving professional, personal, and social qualifications.
- Understanding the need and knowing the possibilities for continuous self-improvement.
- Being aware of the importance and understanding the non-technical aspects and consequences of the activities of an IT engineer.

Program Content

1. Definition of databases and Database Management Systems (DBMS). Areas of application of database technology. Web-based databases. Global database services: library, search, and scientific databases.
2. Data models. Relational data model. Relations, relation schemas, attributes, and keys. Database schema.
3. Reducing redundancy. Normalization of relations. Examples of practical databases.
4. Query languages: QBE (Query by Example) and SQL (Structured Query Language). Selected DBMS implementations for developing practical web-based databases.

Teaching Methods

- Lectures
- Project-based methods
- Laboratory exercises

Education verification method: exam, activity during classes

Online data base- lecture- project

Aims of the Subject

- Conveying knowledge in the subject area.
- Achieving the ability to design a practical relational database.
- Gaining proficiency in using a selected Database Management System (DBMS) for implementing a web-based practical database.

Learning Outcomes **2**

Knowledge

- Gaining knowledge in database technology and database management systems on the global network.
- Understanding and knowing selected issues in data modeling.
- Knowing and understanding tools and data analysis languages (including SQL).

Skills

- Achieving the ability to design the structure of a practical relational database and selecting and using a database management system.
- Utilizing knowledge to solve complex and unusual IT problems.
- Using available literature sources for formulating and solving IT problems.

Social Competencies

- Understanding the need and knowing the possibilities of continuously improving professional, personal, and social qualifications.
- Understanding the need and knowing the possibilities for continuous self-improvement.
- Being aware of the importance and understanding the non-technical aspects and consequences of the activities of an IT engineer.

Program Content

1. Definition of databases and Database Management Systems (DBMS). Areas of application of database technology. Web-based databases. Global database services: library, search, and scientific databases.
2. Data models. Relational data model. Relations, relation schemas, attributes, and keys. Database schema.
3. Reducing redundancy. Normalization of relations. Examples of practical databases.
4. Query languages: QBE (Query by Example) and SQL (Structured Query Language). Selected DBMS implementations for developing practical web-based databases.

Teaching Methods

- Lectures
- Project-based methods
- Laboratory exercises

Education verification method: exam, activity during classes

Web technologies- project

Aims of the Subject

- Conveying knowledge in the subject area.
- Achieving the ability to design the functional structure of a web service.
- Gaining the capability to utilize web services and tools using selected software.

Learning Outcomes

Knowledge

- Gaining knowledge in using web services/tools on the global web with selected software.
- Understanding and knowing methods for analyzing the subject matter.
- Knowing and understanding selected issues in modeling methods.

Skills

- Achieving the ability to design the functional and informational structure of a web page/service.
- Using knowledge to solve complex and atypical IT problems.
- Utilizing available literature sources for formulating and solving IT problems.

Social Competencies

- Demonstrating ongoing professional, personal, and social skill development.
- Understanding the need for and knowing the possibilities of continuously improving one's competencies.
- Being aware of and understanding the non-technical aspects and consequences of an IT engineer's work.

Program Content

1. **Description of the World Wide Web (WWW):**
   - Definition and typology of technology.
   - Examples of web services: email, FTP, information services, Internet TV and radio, social networking sites, web browsers, and internet communicators.
2. **SGML - Standard Generalized Markup Language:**
   - Selected subsets: HTML, VRML, WOML, MathML.
   - HTML – Hypertext Markup Language.
   - Data transmission protocols.
3. **Technical Conditions:**
   - Requirements (computer, peripherals, etc.) for listening to music (or internet radio) or watching TV streams (sports games or other broadcasts).
4. **Web Services and Programming Languages:**
   - Examples: Library of Congress (LoC) web service, Electronic Data Interchange (EDI), e-shopping, online bookstores like Amazon.com.
   - Useful programming languages for building web pages.
   - Web services for e-government and e-administration.

Teaching Methods

- Lectures

- Project-based methods
- Laboratory exercises

Education verification method: exam, activity during classes

Web technologies- tutorials

Aims of the Subject

- Conveying knowledge in the subject area.
- Achieving the ability to design the functional structure of a web service.
- Gaining the capability to utilize web services and tools using selected software.

Learning Outcomes

Knowledge

- Gaining knowledge in using web services/tools on the global web with selected software.
- Understanding and knowing methods for analyzing the subject matter.
- Knowing and understanding selected issues in modeling methods.

Skills

- Achieving the ability to design the functional and informational structure of a web page/service.
- Using knowledge to solve complex and atypical IT problems.
- Utilizing available literature sources for formulating and solving IT problems.

Social Competencies

- Demonstrating ongoing professional, personal, and social skill development.
- Understanding the need for and knowing the possibilities of continuously improving one's competencies.
- Being aware of and understanding the non-technical aspects and consequences of an IT engineer's work.

Program Content

1. **Description of the World Wide Web (WWW):**
   - Definition and typology of technology.
   - Examples of web services: email, FTP, information services, Internet TV and radio, social networking sites, web browsers, and internet communicators.
2. **SGML - Standard Generalized Markup Language:**
   - Selected subsets: HTML, VRML, WOML, MathML.
   - HTML – Hypertext Markup Language.
   - Data transmission protocols.

3. **Technical Conditions:**
    - Requirements (computer, peripherals, etc.) for listening to music (or internet radio) or watching TV streams (sports games or other broadcasts).
4. **Web Services and Programming Languages:**
    - Examples: Library of Congress (LoC) web service, Electronic Data Interchange (EDI), e-shopping, online bookstores like Amazon.com.
    - Useful programming languages for building web pages.
    - Web services for e-government and e-administration.

Teaching Methods

- Lectures
- Project-based methods
- Laboratory exercises

Education verification method: exam, activity during classes

XML technologies- lecture

Aims of the Subject

- Convey knowledge about the subject area.
- Achieve the ability to design the structure of XML documents.
- Gain the capability to utilize XML standards in Electronic Data Interchange (EDI) systems and in the World Wide Web (WWW).

Learning Outcomes

Knowledge

- Gain knowledge in the use of XML standards and services on the global web with selected software.
- Understand and know methods for analyzing the subject matter.
- Know and understand selected issues related to modeling methods. **1**

Skills

- Achieve the ability to design the structure of XML documents for data exchange in EDI systems and web services.
- Use knowledge to solve complex and atypical IT problems.
- Utilize available literature sources for formulating and solving IT problems.

Social Competencies

- Understand the need for and know the possibilities of continuous improvement of professional, personal, and social skills.
- Understand the need for and know the possibilities of continuous improvement of personal competencies.
- Be aware of and understand the non-technical aspects and

consequences of an IT engineer's work.

Program Content

1. **Definition of Document Description Language – XML (eXtensible Markup Language):**
   - The origin and development of XML standards, including applications in World Wide Web services.
   - Examples of XML-defined forms in information services, library services, and data transmission.
2. **SGML – Standard Generalized Markup Language and XML Subset:**
   - Basic terms, user-defined tags, namespaces, Processing Instructions (PI) in XML documents.
   - Characteristics of XML Schema.
3. **Software for Processing XML Documents:**
   - Parsers and XML code readers.
   - XML document formatting.
   - XSL/XSLT (eXtensible Stylesheet Language / XSL Transformation).
   - Query languages related to XML, such as XIRQL (eXtensible Information Retrieval Query Language).
4. **Using XML in Electronic Data Interchange (EDI) Systems:**
   - Applications in e-business, e-commerce, e-finance, e-banking, e-education.
   - Use in CMS/LMS (Content Management System / Learning Management System) services.

Teaching Methods

- Lectures
- Project-based methods
- Laboratory exercises

Education verification method: exam, activity during classes

XML technologies- workshops

Aims of the Subject

- Convey knowledge about the subject area.
- Achieve the ability to design the structure of XML documents.
- Gain the capability to utilize XML standards in Electronic Data Interchange (EDI) systems and in the World Wide Web (WWW).

Learning Outcomes

Knowledge

- Gain knowledge in the use of XML standards and services on the global web with selected software.
- Understand and know methods for analyzing the subject matter.

- Know and understand selected issues related to modeling methods.

Skills

- Achieve the ability to design the structure of XML documents for data exchange in EDI systems and web services.
- Use knowledge to solve complex and atypical IT problems.
- Utilize available literature sources for formulating and solving IT problems.

Social Competencies

- Understand the need for and know the possibilities of continuous improvement of professional, personal, and social skills.
- Understand the need for and know the possibilities of continuous improvement of personal competencies.
- Be aware of and understand the non-technical aspects and consequences of an IT engineer's work.

Program Content

1. **Definition of Document Description Language – XML (eXtensible Markup Language):**
   - The origin and development of XML standards, including applications in World Wide Web services.
   - Examples of XML-defined forms in information services, library services, and data transmission.
2. **SGML – Standard Generalized Markup Language and XML Subset:**
   - Basic terms, user-defined tags, namespaces, Processing Instructions (PI) in XML documents.
   - Characteristics of XML Schema.
3. **Software for Processing XML Documents:**
   - Parsers and XML code readers.
   - XML document formatting.
   - XSL/XSLT (eXtensible Stylesheet Language / XSL Transformation).
   - Query languages related to XML, such as XIRQL (eXtensible Information Retrieval Query Language).
4. **Using XML in Electronic Data Interchange (EDI) Systems:**
   - Applications in e-business, e-commerce, e-finance, e-banking, e-education.
   - Use in CMS/LMS (Content Management System / Learning Management System) services.

Teaching Methods

- Lectures
- Project-based methods
- Laboratory exercises

Education verification method: exam, activity during classes

Software engineering- lecture

Aims of the Subject

- Introduce students to issues, models, and stages of software development, including methodologies and supporting tools.
- Develop skills in specifying and formalizing software requirements, creating software models/designs using various methodologies (both structural and object-oriented) and supporting tools, designing interfaces, and participating in the processes of implementation, testing, validation, and software development.

Learning Outcomes

Knowledge

- Understand and select an appropriate software lifecycle model for a given application.
- Formulate tasks for each stage of the software lifecycle.
- Describe system requirements using formal techniques.
- Analyze the structure of the designed application.
- Choose suitable methods for modeling applications.
- Propose a data coding system.
- Select a software quality assurance system.

2

Skills

- Analyze real-world systems for IT support.
- Formalize the informational content of documents.
- Develop a set of requirements for information systems.
- Design functional and informational structures of information systems using various techniques.
- Create application models using different modeling techniques.
- Apply principles of human-computer interface design and use appropriate application testing techniques.

Social Competencies

- Demonstrate creativity in analyzing and designing applications.
- Identify relationships between different phases of application development.
- Share ideas with others.
- Understand non-technical aspects and consequences of an IT engineer's work, including its impact on the environment and associated responsibilities.

Program Content

1. **Typical Stages of Software Development and Their Content:**
   - Software development models (waterfall, evolutionary, iterative, agile, XP).
   - Prototyping methods.
2. **Analysis of Information and Decision Systems:**
   - Modeling business processes and specifying document content.
   - BPMN and BNF notations.
3. **Types of Requirements:**
   - Requirement acquisition, consolidation, and documentation.
   - Preparation of software specifications according to IEEE standards and acceptance criteria.
   - Formalization notations (templates, scenarios, use cases, hierarchical lists).
   - Requirements management.
4. **Software Design/Modeling:**
   - Structural and object-oriented methodologies.
   - Notations.
5. **System Architecture Design:**
   - Overview of contemporary architectures.
6. **Detailed Techniques for Modeling Processes and Data Structures:**
   - Conceptual, logical, and implementation models.
   - Model mapping.
7. **CASE Environments:**
   - Role of data dictionaries and repositories.
   - Techniques for working with CASE tools.
   - Integrated software development environments.
8. **Standards and User Interface Design:**
   - Principles of designing a proper interface.
   - Supporting tools.
   - Internationalization of interfaces - issues and methods.
9. **Data Coding:**
   - Types and principles.
   - Code construction.
   - Check digits.
10. **Software Testing and Validation:**
    - Objectives, scope, and types of testing methods.
    - Organization of the testing process.
11. **Software Development During the Maintenance Phase:**
    - Process of making changes to software.
    - Configuration management.
12. **Software Quality Assurance Systems:**
    - TQM, ISO 9000x, CMM, and EFQM models.

Teaching Methods

- Problem-based lectures, multimedia techniques, thematic discussions.
- Projects, thematic discussions.

Education verification method: exam, activity during classes

Software engineering- workshop

Aims of the Subject

- Introduce students to issues, models, and stages of software development, including methodologies and supporting tools.
- Develop skills in specifying and formalizing software requirements, creating software models/designs using various methodologies (both structural and object-oriented) and supporting tools, designing interfaces, and participating in the processes of implementation, testing, validation, and software development.

Learning Outcomes

- **Knowledge**
    - Understand and select an appropriate software lifecycle model for a given application.
    - Formulate tasks for each stage of the software lifecycle.
    - Describe system requirements using formal techniques.
    - Analyze the structure of the designed application.
    - Choose suitable methods for modeling applications.
    - Propose a data coding system. **2**
    - Select a software quality assurance system.
- **Skills**
    - Analyze real-world systems for IT support.
    - Formalize the informational content of documents.
    - Develop a set of requirements for information systems.
    - Design functional and informational structures of information systems using various techniques.
    - Create application models using different modeling techniques.
    - Apply principles of human-computer interface design and use appropriate application testing techniques.
- **Social Competencies**
    - Demonstrate creativity in analyzing and designing applications.
    - Identify relationships between different phases of application development.
    - Share ideas with others.
    - Understand non-technical aspects and consequences of an IT engineer's work, including its impact on the environment and associated responsibilities.

Program Content

Lectures

1. **Typical Stages of Software Development and Their Content:**
   - Software development models (waterfall, evolutionary, iterative, agile, XP).
   - Prototyping methods.
2. **Analysis of Information and Decision Systems:**
   - Modeling business processes and specifying document content.
   - BPMN and BNF notations.
3. **Types of Requirements:**
   - Requirement acquisition, consolidation, and documentation.
   - Preparation of software specifications according to IEEE standards and acceptance criteria.
   - Formalization notations (templates, scenarios, use cases, hierarchical lists).
   - Requirements management.
4. **Software Design/Modeling:**
   - Structural and object-oriented methodologies.
   - Notations.
5. **System Architecture Design:**
   - Overview of contemporary architectures.
6. **Detailed Techniques for Modeling Processes and Data Structures:**
   - Conceptual, logical, and implementation models.
   - Model mapping.
7. **CASE Environments:**
   - Role of data dictionaries and repositories.
   - Techniques for working with CASE tools.
   - Integrated software development environments.
8. **Standards and User Interface Design:**
   - Principles of designing a proper interface.
   - Supporting tools.
   - Internationalization of interfaces - issues and methods.
9. **Data Coding:**
   - Types and principles.
   - Code construction.
   - Check digits.
10. **Software Testing and Validation:**
    - Objectives, scope, and types of testing methods.
    - Organization of the testing process.
11. **Software Development During the Maintenance Phase:**
    - Process of making changes to software.
    - Configuration management.
12. **Software Quality Assurance Systems:**
    - TQM, ISO 9000x, CMM, and EFQM models.

Project

1. **Diagram Editors (e.g., MS Visio) and Working Principles:**
   - Using diagramming tools.
2. **Analysis - Description of Client Actions in a Real System:**

    - Analyzing client interactions.
3. **Analysis - Modeling Business Processes:**
    - Creating business process models.
4. **Analysis - Document Content and Formatting Analysis, BNF Notation:**
    - Analyzing document content and structure.
5. **Requirements - Modeling Requirements, Functional Trees, Context Diagrams:**
    - Creating requirement models and diagrams.
6. **CASE Tools - Working Principles:**
    - Using CASE tools effectively.
7. **Modeling Functional Structure of Applications:**
    - Designing application functionality.
8. **Data Modeling - ERD Diagrams:**
    - Creating Entity-Relationship Diagrams.
9. **Designing Data Coding Systems:**
    - Developing data coding strategies.
10. **Interface Design:**
    - Designing user interfaces.

Teaching Methods

- Problem-based lectures, multimedia techniques, thematic discussions.
- Projects, thematic discussions.

Education verification method: exam, activity during classes

Awareness of the decisions' value- workshop

Course Objectives

- Prepare students to analyze their own hierarchy of values.
- Demonstrate the relationship between the choices they make and their needs and values.
- Enhance students' ability to make independent and conscious decisions.

Learning Outcomes

In terms of knowledge:

- Students define and list the stages of decision-making.
- Students explain the specifics of individual and group decision-making.

In terms of skills:

- Students effectively use axiological concepts.
- Students analyze and verify their hierarchy of values.
- Students improve their ability to make decisions (both individually and in groups) based on their values.

- Students analyze different decision-making styles.

In terms of social competencies:

- Students justify their views and reasoning.
- Students demonstrate responsibility for their choices.
- Students maintain openness to different decision-making styles.
- Students focus on the needs of others when making decisions.

Course Content

1. Perception of values in selected philosophical currents (relativism, absolutism, objectivism, subjectivism).
2. Real versus declared values.
3. Selected axiological problems of the modern world.
4. Perception of good and evil.
5. Awareness of the current hierarchy of values.
6. Conflict of values.
7. Freedom and responsibility in action.
8. Goals and values in life planning – the significance of having a goal in life.

Teaching Methods

- Didactic discussion
- Brainstorming
- Dramatic techniques
- Workshop method

Education verification method: exam, activity during classes