Akademia Humanistyczno Ekonomiczna w Łodzi

ERASMUS+ INCOMING STUDENTS
ACADEMIC OFFER - COURSES TAUGHT IN ENGLISH
COMPUTER SCIENCE
SPRING 2025/2026

BACHELOR

COMPUTER SCIENCE

SUBJECTS LISTED ARE A CONTINUATION OF COURSES AND REQUIRE INTERESTED STUDENTS TO HAVE KNOWLEDGE IN THE FIELDS IN ORDER TO BE ABLE TO PARTICIPATE

1st year, 2nd semester

Discrete mathematics – lecture

3 ECTS

The purpose of the course is to familiarize the student with combinatorics along with the necessary elements of algebra and number theory. Introduction to graph theory and its applications. The aim of the subject is to introduce students to combinatorics, including essential elements of algebra and number theory. It also provides an introduction to graph theory and its applications.

Learning Outcomes

- **In Terms of Knowledge:** Defines basic theorems of number theory. Understands and applies the concepts of recursion and induction, and is able to use reasoning rules and conduct simple proofs. Defines fundamental theorems of combinatorics. Formulates and simplifies practical problems using mathematical language.
- **In Terms of Skills:** Solves recurrence equations. Utilizes the principle of mathematical induction. Performs calculations in matrix arithmetic. Formulates and simplifies practical problems using mathematical language.
- **In Terms of Social Competencies:** Demonstrates creative and inventive thinking. Understands the need and recognizes the opportunities for continuous improvement of personal competencies.

Course Content

1. Introductory concepts and relations.
2. Introduction to graph theory.
3. Graphs and trees.
4. Induction and recursion.
5. Elements of combinatorics.
6. Basic counting methods.

Teaching Methods: Lecture (partially using multimedia tools), Didactic discussion, Classical problem-based method, Project-based method.

Education verification method: exam, activity during classes

Discrete mathematics – project

4 ECTS

The purpose of the course is to familiarize the student with combinatorics along with the necessary elements of algebra and number theory. Introduction to graph theory and its applications. The aim of the subject is to introduce students to combinatorics, including essential elements of algebra and number theory. It also provides an introduction to graph theory and its applications.

Learning Outcomes

- **In Terms of Knowledge:** Defines basic theorems of number theory. Understands and applies the concepts of recursion and induction, and is able to use reasoning rules and conduct simple proofs. Defines fundamental theorems of combinatorics. Formulates and simplifies practical problems using mathematical language.
- **In Terms of Skills:** Solves recurrence equations. Utilizes the principle of mathematical induction. Performs calculations in matrix arithmetic. Formulates and simplifies practical problems using mathematical language.
- **In Terms of Social Competencies:** Demonstrates creative and inventive thinking. Understands the need and recognizes the opportunities for continuous improvement of personal competencies.

Course Content

1. Introductory concepts and relations.
2. Introduction to graph theory.
3. Graphs and trees.
4. Induction and recursion.
5. Elements of combinatorics.
6. Basic counting methods.

ZDW: Operating Systems / Systems for Managing Computer Hardware and Applications – workshops — 2 ECTS

The aims of the subjects are understanding fundamental concepts related to operating systems; familiarizing students with the role of the operating system in the functioning of a computer system; introducing students to the most important operating systems.

Learning Outcomes:

- **In terms of knowledge:** Defines fundamental concepts related to operating systems. Can explain the role of the operating system in the functioning of a computer system. Chooses appropriate system solutions for specified goals.
- **In terms of skills:** Acquires practical skills in managing various operating systems. Gains practical skills in managing operational memory. Develops practical skills in administering file systems, virtual memory, and dynamic libraries.
- **In terms of social competences:** Demonstrates the ability for continuous learning, improving, and enhancing professional, personal, and social competencies.

Course Content:

1. Structure and functions of an operating system.
2. Concept of processes and threads. Processor time allocation.
3. Management of operational memory allocation.
4. Virtual memory mechanisms.
5. File systems.
6. Input-output system handling.
7. Working in networks.
8. User accounts in operating systems. Access control.
9. Mechanism of dynamic libraries.
10. System services/daemons.

Teaching Methods: Lecture, Classical problem-based method, Project-based method

ZDW: Operating Systems / Systems for Managing Computer Hardware and Applications — 2 ECTS

The aims of the subject are understanding fundamental concepts related to operating systems; familiarizing students with the role of the operating system in the functioning of a computer system; introducing students to the most important operating systems.

Learning Outcomes:

- **In terms of knowledge:** Defines fundamental concepts related to operating systems. Can explain the role of the operating system in the functioning of a computer system. Chooses appropriate system solutions for specified goals.
- **In terms of skills:** Acquires practical skills in managing various operating systems. Gains practical skills in managing operational memory. Develops practical skills in administering file systems, virtual memory, and dynamic libraries.
- **In terms of social competencies:** Demonstrates the ability for continuous learning, improving, and enhancing professional, personal, and social competencies.

Course Content:

1. Structure and functions of an operating system.
2. Concept of processes and threads. Processor time allocation.
3. Management of operational memory allocation.
4. Virtual memory mechanisms.
5. File systems.
6. Input-output system handling.
7. Working in networks.
8. User accounts in operating systems. Access control.
9. Mechanism of dynamic libraries.
10. System services/daemons.

Teaching Methods: Lecture, Classical problem-based method, Project-based method

Education verification method: exam, activity during classes

ZDW: Operating Systems / Systems for Managing Computer Hardware and Applications – project — 3 ECTS

The aims of the subject are understanding fundamental concepts related to operating systems; familiarizing students with the role of the operating system in the functioning of a computer system; introducing students to the most important operating systems.

Learning Outcomes:

- **In terms of knowledge:** Defines fundamental concepts related to operating systems. Can explain the role of the operating system in the functioning of a computer system. Chooses appropriate system solutions for specified goals.
- **In terms of skills:** Acquires practical skills in managing various operating systems. Gains practical skills in managing operational memory. Develops practical skills in administering file systems, virtual memory, and dynamic libraries.
- **In terms of social competencies:** Demonstrates the ability for continuous learning, improving, and enhancing professional, personal, and social competencies.

Course Content:

1. Structure and functions of an operating system.
2. Concept of processes and threads. Processor time allocation.
3. Management of operational memory allocation.
4. Virtual memory mechanisms.
5. File systems.
6. Input-output system handling.
7. Working in networks.
8. User accounts in operating systems. Access control.
9. Mechanism of dynamic libraries.
10. System services/daemons.

Teaching Methods: Lecture, Classical problem-based method, Project-based method

Education verification method: exam, activity during classes

Basics of programming 2 – lecture

2 ECTS

The aims of the subject are to impart knowledge in the subject area; to enhance the ability to design computer courses and create efficient source code.

Learning Outcomes:

- **In terms of knowledge:** Acquires knowledge in the area covered by the subject, including the construction and development of computer courses. Understands and can explain methods for analyzing the subject matter. Understands and can explain selected topics related to modeling methods.
- **In terms of skills:** Has achieved and refined the ability to design efficient source code for computer courses. Can apply knowledge to solve complex and non-standard IT problems. Can use available literature to formulate and solve IT problems.
- **In terms of social competencies:** Understands the need and opportunities for continuous improvement of professional, personal, and social qualifications. Recognizes the need and knows how to continuously enhance personal competencies. Is aware of and understands the non-technical aspects and consequences of the work of a computer engineer.

Course Content:

1. Definition and typology of programming languages, including an overview and examples. General structure and methods of code translation for computer courses.
2. Course construction, covering data types such as basic and complex types, constants, variables, arrays, as well as record and file types.
3. File operations including opening and closing files, writing to disk, arithmetic and logical operators, assignment and conditional statements, and macro substitution.
4. Refinement of iterative operations in course construction, with elements of object-oriented programming, including classes, objects, properties, and methods for handling objects.

Teaching Methods: Lectures, Project-based methods, Auditory/laboratory exercises

Education verification method: exam, activity during classes

Basics of programming 2 – exercises

2 ECTS

The aims of the subject are to impart knowledge in the subject area; to enhance the ability to design computer courses and create efficient source code.

Learning Outcomes:

- **In terms of knowledge:** Acquires knowledge in the area covered by the subject, including the construction and development of computer courses. Understands and can explain methods for analyzing the subject matter. Understands and can explain selected topics related to modeling methods.
- **In terms of skills:** Has achieved and refined the ability to design efficient source code for computer courses. Can apply knowledge to solve complex and non-standard IT problems. Can use available literature to formulate and solve IT problems.
- **In terms of social competences:** Understands the need and opportunities for continuous improvement of professional, personal, and social qualifications. Recognizes the need and knows how to continuously enhance personal competencies. Is aware of and understands the non-technical aspects and consequences of the work of a computer engineer.

Course Content:

1. Definition and typology of programming languages, including an overview and examples. General structure and methods of code translation for computer courses.
2. Course construction, covering data types such as basic and complex types, constants, variables, arrays, as well as record and file types.
3. File operations including opening and closing files, writing to disk, arithmetic and logical operators, assignment and conditional statements, and macro substitution.
4. Refinement of iterative operations in course construction, with elements of object-oriented programming, including classes, objects, properties, and methods for handling objects.

Teaching Methods: Lectures, Project-based methods, Auditory/laboratory exercises

Education verification method: exam, activity during classes

Basics of programming 2 – project

2 ECTS

The aims of the subject are to impart knowledge in the subject area; to enhance the ability to design computer courses and create efficient source code.

Learning Outcomes:

- **In terms of knowledge:** Acquires knowledge in the area covered by the subject, including the construction and development of computer courses. Understands and can explain methods for analyzing the subject matter. Understands and can explain selected topics related to modeling methods.
- **In terms of skills:** Has achieved and refined the ability to design efficient source code for computer courses. Can apply knowledge to solve complex and non-standard IT problems. Can use available literature to formulate and solve IT problems.
- **In terms of social competencies:** Understands the need and opportunities for continuous improvement of professional, personal, and social qualifications. Recognizes the need and knows how to continuously enhance personal competencies. Is aware of and understands the non-technical aspects and consequences of the work of a computer engineer.

Course Content:

1. Definition and typology of programming languages, including an overview and examples. General structure and methods of code translation for computer courses.
2. Course construction, covering data types such as basic and complex types, constants, variables, arrays, as well as record and file types.
3. File operations including opening and closing files, writing to disk, arithmetic and logical operators, assignment and conditional statements, and macro substitution.
4. Refinement of iterative operations in course construction, with elements of object-oriented programming, including classes, objects, properties, and methods for handling objects.

Teaching Methods: Lectures, Project-based methods, Auditory/laboratory exercises

Education verification method: exam, activity during classes

Numerical methods in computer science – lecture

2 ECTS

The purpose of the class is to acquire basic knowledge of the effect of floating point arithmetic on the result of numerical calculations, to learn numerically correct algorithms for basic tasks of mathematical analysis and linear algebra. To acquire basic knowledge regarding the impact of floating-point arithmetic on numerical computation results. To learn numerically accurate algorithms for fundamental tasks in mathematical analysis and linear algebra.

Learning Outcomes:

- **In terms of knowledge:** Understands and can explain algorithmic methods for solving optimization problems. Knows and comprehends basic methods for formulating mathematical models for practical and theoretical optimization problems. Understands the principles of formulating and classifying optimization problems.
- **In terms of skills:** Can select appropriate numerical methods for solving optimization problems. Can apply knowledge to solve complex and non-standard optimization problems. Can utilize available literature to formulate and solve IT problems.
- **In terms of social competencies:** Understands the need and opportunities for continuous improvement of professional, personal, and social qualifications. Recognizes the need and knows how to continuously enhance personal competencies. Can engage in professional discussions with users of numerical methods in practice.

Course Content:

1. Solving systems of linear equations, including Gaussian elimination (review), LU decomposition (Doolittle's algorithm), Cholesky decomposition, and issues with numerically ill-conditioned matrices.
2. Techniques for sparse matrices, methods of representing sparse matrices, optimal ordering of equations, Minimum Degree method, elimination trees, supernodes, factorization, and interior-point methods.
3. Iterative methods for solving systems of equations, including classic methods (Jacobi method, Gauss-Seidel method, SOR method), and projection methods (conjugate gradients method, Arnoldi method, GMRES method).
4. Mathematical formulation of optimization problems, specific and general optimization criteria, and objective functions. Principles of formulating and classifying optimization problems, practical and mathematical formulations, and examples of optimization problems.
5. Methods for nonlinear static optimization problems, optimality conditions for unconstrained and equality-constrained problems, optimality conditions for general nonlinear programming problems (Karush-Kuhn-Tucker conditions), and standard form of nonlinear programming problems.
6. Gradient-free methods: stochastic overview, Nelder-Mead algorithm,

orthogonal direction method with optimal step length selection, Hooke-Jeeves method, Powell method, and Rosenbrock method.
7. Gradient methods: steepest descent method, conjugate gradient method, and variable metric methods (Davidon-Fletcher-Powell, Broyden-Fletcher-Goldfarb-Shanno).
8. Methods for optimizing multidimensional nonlinear problems with constraints, including gradient-based methods with backtracking, projected gradient methods, methods with penalty functions, and sequential quadratic programming methods.

Teaching Methods: lectures, project-based methods, classical problem-based method

Education verification method: exam, activity during classes

Numerical methods in computer science – exercises

2 ECTS

The purpose of the class is to acquire basic knowledge of the effect of floating point arithmetic on the result of numerical calculations, to learn numerically correct algorithms for basic tasks of mathematical analysis and linear algebra. To acquire basic knowledge regarding the impact of floating-point arithmetic on numerical computation results. To learn numerically accurate algorithms for fundamental tasks in mathematical analysis and linear algebra.

Learning Outcomes:

- **In terms of knowledge:** Understands and can explain algorithmic methods for solving optimization problems. Knows and comprehends basic methods for formulating mathematical models for practical and theoretical optimization problems. Understands the principles of formulating and classifying optimization problems.
- **In terms of skills:** Can select appropriate numerical methods for solving optimization problems. Can apply knowledge to solve complex and non-standard optimization problems. Can utilize available literature to formulate and solve IT problems.
- **In terms of social competencies:** Understands the need and opportunities for continuous improvement of professional, personal, and social qualifications. Recognizes the need and knows how to continuously enhance personal competencies. Can engage in professional discussions with users of numerical methods in practice.

Course Content:

1. Solving systems of linear equations, including Gaussian elimination (review), LU decomposition (Doolittle's algorithm), Cholesky decomposition, and issues with numerically ill-conditioned matrices.
2. Techniques for sparse matrices, methods of representing sparse

matrices, optimal ordering of equations, Minimum Degree method, elimination trees, supernodes, factorization, and interior-point methods.

3. Iterative methods for solving systems of equations, including classic methods (Jacobi method, Gauss-Seidel method, SOR method), and projection methods (conjugate gradients method, Arnoldi method, GMRES method).
4. Mathematical formulation of optimization problems, specific and general optimization criteria, and objective functions. Principles of formulating and classifying optimization problems, practical and mathematical formulations, and examples of optimization problems.
5. Methods for nonlinear static optimization problems, optimality conditions for unconstrained and equality-constrained problems, optimality conditions for general nonlinear programming problems (Karush-Kuhn-Tucker conditions), and standard form of nonlinear programming problems.
6. Gradient-free methods: stochastic overview, Nelder-Mead algorithm, orthogonal direction method with optimal step length selection, Hooke-Jeeves method, Powell method, and Rosenbrock method.
7. Gradient methods: steepest descent method, conjugate gradient method, and variable metric methods (Davidon-Fletcher-Powell, Broyden-Fletcher-Goldfarb-Shanno).
8. Methods for optimizing multidimensional nonlinear problems with constraints, including gradient-based methods with backtracking, projected gradient methods, methods with penalty functions, and sequential quadratic programming methods.

Teaching Methods: Lectures, Project-based methods, Classical problem-based methods

Education verification method: exam, activity during classes

Numerical methods in computer science – project

2 ECTS

The purpose of the class is to acquire basic knowledge of the effect of floating point arithmetic on the result of numerical calculations, to learn numerically correct algorithms for basic tasks of mathematical analysis and linear algebra. To acquire basic knowledge regarding the impact of floating-point arithmetic on numerical computation results. To learn numerically accurate algorithms for fundamental tasks in mathematical analysis and linear algebra.

Learning Outcomes:

- **In terms of knowledge:** Understands and can explain algorithmic methods for solving optimization problems. Knows and comprehends basic methods for formulating mathematical models for practical and theoretical optimization problems. Understands the principles of formulating and classifying optimization problems.
- **In terms of skills:** Can select appropriate numerical methods for solving optimization problems. Can apply knowledge to solve complex and non-standard optimization problems. Can utilize available literature to formulate and solve IT problems.
- **In terms of social competences:** Understands the need and opportunities for continuous improvement of professional, personal, and social qualifications. Recognizes the need and knows how to continuously enhance personal competencies. Can engage in professional discussions with users of numerical methods in practice.

Course Content:

1. Solving systems of linear equations, including Gaussian elimination (review), LU decomposition (Doolittle's algorithm), Cholesky decomposition, and issues with numerically ill-conditioned matrices.
2. Techniques for sparse matrices, methods of representing sparse matrices, optimal ordering of equations, Minimum Degree method, elimination trees, supernodes, factorization, and interior-point methods.
3. Iterative methods for solving systems of equations, including classic methods (Jacobi method, Gauss-Seidel method, SOR method), and projection methods (conjugate gradients method, Arnoldi method, GMRES method).
4. Mathematical formulation of optimization problems, specific and general optimization criteria, and objective functions. Principles of formulating and classifying optimization problems, practical and mathematical formulations, and examples of optimization problems.
5. Methods for nonlinear static optimization problems, optimality conditions for unconstrained and equality-constrained problems, optimality conditions for general nonlinear programming problems (Karush-Kuhn-Tucker conditions), and standard form of nonlinear

programming problems.

6. Gradient-free methods: stochastic overview, Nelder-Mead algorithm, orthogonal direction method with optimal step length selection, Hooke-Jeeves method, Powell method, and Rosenbrock method.
7. Gradient methods: steepest descent method, conjugate gradient method, and variable metric methods (Davidon-Fletcher-Powell, Broyden-Fletcher-Goldfarb-Shanno).
8. Methods for optimizing multidimensional nonlinear problems with constraints, including gradient-based methods with backtracking, projected gradient methods, methods with penalty functions, and sequential quadratic programming methods.

Teaching Methods: Lectures, Project-based methods, Classical problem-based methods

Education verification method: exam, activity during classes

ZDW: Intellectual Property Protection

1 ECTS

To acquire knowledge about intellectual property and its protection in Poland and globally.

Learning Outcomes:

- **In terms of knowledge:** Understands and can explain concepts related to intellectual property, including copyright. Knows the typical relationships between intellectual property protection and fair competition, innovation, and economic growth. Understands the principles of intellectual property protection. Can distinguish between moral rights and economic rights in copyright.
- **In terms of skills:** Can accurately define a work and other intellectual property subjects in legal and economic terms. Can evaluate which works are not protected by copyright and justify why. Can select information and statistical data to analyze the economic impact of intellectual property.
- **In terms of social competencies:** Is aware of the social and economic importance of intellectual property protection. Acts professionally with respect for intellectual property, including creating scholarly texts and simple information without infringing copyright.

Course Content:

1. The historical development of intangible asset protection.
2. International and national aspects of intellectual property protection.
3. The origins and current status of copyright and related rights.
4. The relationship between intellectual property protection and competition policy, combating unemployment, innovation, and economic growth.
5. The subject matter and parties involved in copyright law - definitions.
6. Moral rights of authors concerning protected works.
7. The concept and catalogue of economic rights and exploitation fields of a work. Selected issues related to licensing.
8. Forms of infringement of moral and economic copyright rights, including plagiarism, piracy, and databases. The role of collective rights management organizations.
9. The concept and principles of fair use for private and public purposes. Library and school rights. The right of quotation.
10. Special protection for computer courses, images, and correspondence.
11. Protection of inventions, trademarks, and industrial designs. Community trademarks.
12. Civil and criminal liability principles for intellectual property rights violations.

Communication and relationship building[i]

1 ECTS

Course objectives: The course will explore strategies for establishing and maintaining connections through effective communication. It covers communincation tools such as naming emotions, asking questions, and active listening techniques like paraphrasing and mirroring. The importance of being precise and clear in communication will also be emphasised. Verbal and non-verbal communication means are going to be discovered, discussed and explored together with communication barriers, including cross-cultural context. The course participants will be equipped with practical tools and insights to communicate more effectively, build rapport and trust in order to establish relationships in both personal and professional settings.

Learning outcomes:

- **In terms of knowledge:** The student has knowledge of verbal and non-verbal communication, including methods and styles of communication as well as communication barriers
- **In terms of skills:** The student establishes and deepens relationships with the group. The student chooses an effective communication strategy. The student uses selected tools for effective communication. The student is able to present themselves in an appropriate way to the situation. The student organizes teamwork.
- **In terms of social competences:** The student shows openness to solving individual and group communication problems. The student engages in teamwork and plays various group roles.

Course content:

1. Ways to effectively establish contact with another person.
2. Tools for effective communication: naming feelings, using open questions, active listening: paraphrase, mirroring, precision of the message, “I” message.
3. The role of verbal and non-verbal communication.
4. Communication barriers.
5. The role and importance of emotions in the process of communication and integration.
6. Online communication.

2nd year, 4th semester

ZDW: Ochrona danych/Bezpieczeństwo danych / ZDW: Data protection/Data security – lecture

1 ECTS

The purpose of the course is to familiarize students with the basic threats to information security in information systems and how to secure them. To familiarize students with the basic threats to information security in information systems and methods of securing them.

Learning Outcomes:

- **In Terms of Knowledge:** Students will: Gain knowledge about problems related to threats in information systems. Understand the organization of information security systems within an organization. Know and understand the significance of standards in information security and their contents.
- **In Terms of Skills:** Students will be able to: Design the user structure of a relational database and select and utilize a database management system. Apply knowledge to solve complex and atypical issues related to data access protection, including identification, authorization, and permissions. Utilize available literature to formulate and solve problems related to data protection.
- **In Terms of Social Competencies:** Students will: Recognize the need and possibilities for continuous improvement of professional, personal, and social skills. Understand the necessity of continuous self-improvement in competencies. Be aware of the importance of and understand non-technical aspects and consequences of an engineer's activities concerning ensuring hardware continuity—identifying threats.

Course Content:

1. Types of threats to information systems and their impact. Methods for classifying threats. CIA Triad (Confidentiality, Integrity, Availability).
2. The concept of system security. Responsibility for security. Organizational and legal issues.
3. Protection of devices, applications, and data: methods and scope.
4. Issues related to data protection. Identifying individuals violating security. Methods of malicious interference. Incident registration.
5. IT methods for data access protection (identification, authorization, accounting). Granting permissions.
6. Methods for securing data against loss. Backup creation and storage principles.
7. Emergency procedures. Ensuring hardware continuity. Causes of failures—identification and threats.
8. Secure systems: structural aspects, RAID systems.
9. Antivirus protection. Types of viruses and their dissemination methods.

Antivirus courses.

10. Security in local and wide area networks. Security of transmitted data.
11. Cryptography issues: encryption and breaking of security. Steganography. Electronic signatures and their infrastructure.
12. Organizational issues: recovery procedures and plans, emergency centers, outsourcing. Implementing a security system in a company. Security policy.
13. Standards. Security system audit.
14. Security of web applications.

Teaching Methods: Lecture: For presenting theoretical aspects and foundational knowledge; **Conversational Lecture**: To encourage interactive discussion and deeper understanding; **Brainstorming**: For generating ideas and solutions related to security challenges; **Project**: Practical application of concepts through project work to design and implement security measures.

Education verification method: exam, activity during classes

ZDW: Ochrona danych/bezpieczeństwo danych / ZDW: Dataprotection/Data security – project

2 ECTS

The purpose of the course is to familiarize students with the basic threats to information security in information systems and how to secure them. To familiarize students with the basic threats to information security in information systems and methods of securing them.

Learning Outcomes:

- **In Terms of Knowledge:** Students will: Gain knowledge about problems related to threats in information systems. Understand the organization of information security systems within an organization. Know and understand the significance of standards in information security and their contents.
- **In Terms of Skills:** Students will be able to: Design the user structure of a relational database and select and utilize a database management system. Apply knowledge to solve complex and atypical issues related to data access protection, including identification, authorization, and permissions. Utilize available literature to formulate and solve problems related to data protection.
- **In Terms of Social Competencies:** Students will: Recognize the need and possibilities for continuous improvement of professional, personal, and social skills. Understand the necessity of continuous self-improvement in competencies. Be aware of the importance of and understand non-technical aspects and consequences of an engineer's activities concerning ensuring hardware continuity—identifying threats.

Course Content:

1. Types of threats to information systems and their impact. Methods for classifying threats. CIA Triad (Confidentiality, Integrity, Availability).
2. The concept of system security. Responsibility for security. Organizational and legal issues.
3. Protection of devices, applications, and data: methods and scope.
4. Issues related to data protection. Identifying individuals violating security. Methods of malicious interference. Incident registration.
5. IT methods for data access protection (identification, authorization, accounting). Granting permissions.
6. Methods for securing data against loss. Backup creation and storage principles.
7. Emergency procedures. Ensuring hardware continuity. Causes of failures—identification and threats.
8. Secure systems: structural aspects, RAID systems.
9. Antivirus protection. Types of viruses and their dissemination methods. Antivirus programmes.
10. Security in local and wide area networks. Security of transmitted data.
11. Cryptography issues: encryption and breaking of security. Steganography. Electronic signatures and their infrastructure.
12. Organizational issues: recovery procedures and plans, emergency centers, outsourcing. Implementing a security system in a company. Security policy.
13. Standards. Security system audit.
14. Security of web applications.

Teaching Methods: Lecture: For presenting theoretical aspects and foundational knowledge; **Conversational Lecture**: To encourage interactive discussion and deeper understanding; **Brainstorming**: For generating ideas and solutions related to security challenges; **Project**: Practical application of concepts through project work to design and implement security measures.

Education verification method: exam, activity during classes

[i] also offered in Pedagogy