BANCA DE SUBIECTE PENTRU PROGRAMUL DE STUDII
INFORMATICA IN LIMBA ENGLEZA

Anul I			
Nr. crt.	**Categoria formativă**	**Plan invățământ site**	**Subiecte**
1.	DC	**Mathematical Analysis**	1. Infimum and Supremum: definition, examples, characterization by sequences. 2. Countable sets: definition, examples. Z and Q are countable sets (proof) 3. Convergent sequences, Cauchy sequences. Definition, examples and counterexamples. 4. Stolz-Cesaro lemma and consequences. (proof+applications) 5. Metric spaces (definition, examples). Convergent sequences in metric spaces 6. Existence of the limit of a real function at a point. Definition, characterization by sequences. Examples 7. Continuous functions. Definition, examples. 8. Differentiable functions. Definition, examples. Connection with continuous functions 9. Numerical series. Definition of series's convergence, examples. 10. Generalized harmonic series (proof for convergence+applications) 11. Taylor series. MacLaurin series. Definition, applications.
2.	DF	**Data Structures**	**Data Structures** 1. One-dimensional and two-dimensional vectors: definition, operations, practical use example 2. Singly linked list: definition, operations of adding, traversing, deleting elements

			3. Doubly linked list: definition, operations of adding, traversing, deleting elements 4. Binary search trees: definition, traversing, searching in trees 5. Binary trees associated with an arithmetic expression: generation, example of use
3.	DF	**Fundamentals of Programming**	**Fundamentals of programming** 1. Operators in C language (types, semantics, precedence and associativity rules, examples). 2. Decisional instructions in C language (syntax, semantics, examples). 3. Repetitive instructions in C language (syntax, semantics, examples). 4. User-defined functions in C language (declaration, implementation, call, parameter transfer, functions with simple parameters/data arrays). 5. Data arrays (arrays) in C language (declaration, initialization, accessing elements, sending as parameters in functions). 6. Fundamental algorithms for working with data arrays, implemented in C language (finding minimum/maximum, number of occurrences of an element, frequency vector, sequential search, bubble or selection sorting). 7. Pointers (definition, declaration, initialization, connection between pointers and arrays/data arrays, functions with pointer parameters). 8. Dynamic memory management (heap) in the C language (specific functions, examples). 9. Structures in the C language (definition, declaration, accessing the fields of a structure, nested structures, arrays of structures). 10. Functions with structures (functions that receive a structure parameter, functions that return structures, functions with pointers to structures parameters).

4.	DC	**Linear Algebra**	**Linear Algebra** 1. Determinant of a matrix: definition, fundamental properties. 2. Inverse of a matrix: definition, existence and uniqueness, adjoint matrix method and Gauss-Jordan method for computing the inverse matrix. 3. Vector spaces: definition, examples, fundamental properties. 4. Vector subspace: definition, examples, subspace criteria. 5. Basis of a finitely generated vector space. Coordinates of a vector in a given basis of the space. Dimension of a vector space. 6. Linear transformations: definition, examples, fundamental properties, matrix associated with a linear transformation. 7. Kernel and image of a linear transformation: definitions, properties, relationship between dimensions (Rank-Defect Theorem). 8. Eigenvalues and eigenvectors: definition, properties. 9. Diagonalization of a matrix: criteria for diagonalizability. 10. Positive definite matrices, positive definite quadratic forms: definition, examples, fundamental properties.
5.	DF	**Computer Architectures**	**Computer Architectures** 1. The virtual layers of a computing system (abstraction, layering, virtual machine, characteristics of each layer) 2. Central processing unit (CPU): components, characteristics, registers, instruction cycle, segmentation (general, case study: Intel 8086 CPU). 3. Computer memory system: types, characteristics, locality of memory references, levels 4. Cache memory: characteristics, cache mapping techniques (definition, examples) 5. IA-32 Architecture: operating mode, registers, segmentation model, translating addresses (without paging)

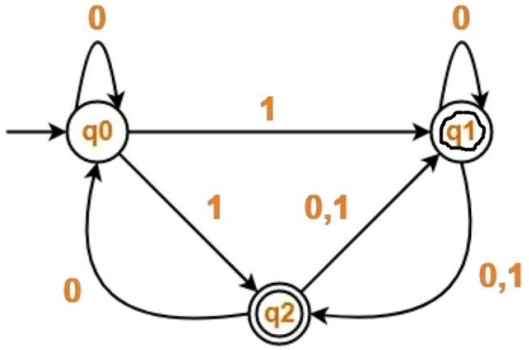
			6. Basics of Virtual Memory (VM): characteristics, dynamic address translation, fast address translation, page replacement 7. IA-32 memory model: segmentation, virtual memory, translating addresses 8. CISC and RISC architectures (main features, advantages and disadvantages) 9. Pipelining: basic idea, performance, hazards (definitions, examples). 10. Parallelism in computing systems: categories of flows, Flynn's taxonomy (definition, examples, analogies)
6.	DF	**Differential and Integral Calculus**	**Differential and Integral Calculus** 1. Fréchet differential of a real valued function of several real variables - definition, properties, examples. 2. Directional derivatives of a real valued function of several real variables. Partial derivatives. 3. Differentiation of composite functions of several real variables. Partial derivatives of composite functions. The chain rule. 4. Higher-order partial derivatives. Schwarz's symmetry theorem. 5. Higher-order differentials for functions of several real variables. Taylor's Formula. 6. Local extrema of functions of several real variables. 7. Conditional extrema of functions of several real variables. 8. Line integrals of the first kind (scalar line integrals)- definition, properties, examples. 9. Line integrals of the second kind (line integrals of vector fields) - definition, properties, examples. Path independent line integrals. 10. Double integrals - definition, properties, change of variables, examples.
7.	DF	**Fundamental Algorithms**	**Fundamental algorithms**

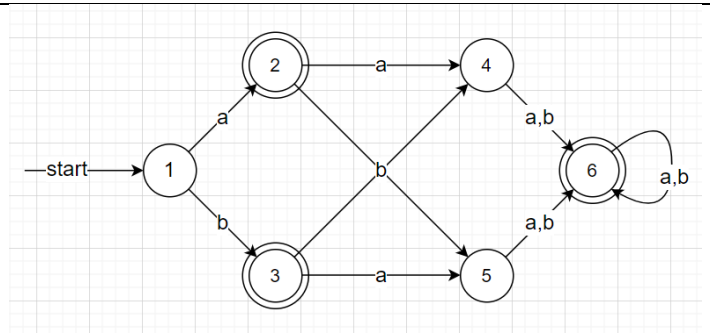
			1. Fundamental algorithms: processing the digits of a number, prime numbers, inversion 2. Selection and insertion sort algorithms: description of algorithms, analysis of their efficiency 3. Sorting algorithms by interleaving and pivoting: description of algorithms, analysis of their efficiency 4. Search algorithms: sequential search and binary search 5. Algorithms for searching for substrings of characters
8.	DF	**Algebraical Fundamentals of Informatics**	1. Lattices: Definition, properties, examples. 2. Groups: Definition, properties, examples. 3. Subgroups: Definition, properties, examples. 4. Subgroups of the group $(\mathbb{Z}, +)$. 5. The order of an element in a finite group. Application: Determine the subgroups of the group $(\mathbb{Z}_{18}, +)$ and the orders of its elements. 6. Rings: Definition, properties, examples. 7. Invertible elements in a ring. Zero divisors. Application: Determine the invertible elements and the zero divisors in the ring $(\mathbb{Z}_{20}, +, *)$. What do you observe? 8. Euclid's Algorithm. Application: Determine the inverse of the element 15 in the ring $(\mathbb{Z}_{1372}, +, *)$ using Euclid's algorithm. 9. Chinese Remainder Theorem. Application: Using the Chinese Remainder Theorem, find the solutions to the system: $x \equiv 7 \mod 8$, $x \equiv 6 \mod 9$, $x \equiv 3 \mod 11$. 10. Affine Cipher. Application: Using the affine cipher with the key (7, 10), encrypt the message "LICENTA", knowing that a 27-letter alphabet is

			used (the regular alphabet without diacritics plus the last symbol as a space). The operations are performed in ($\mathbb{Z}_{27}$, +, *). Find the decryption key.
9.	DC	**Analytical and Differential Geometry**	1. Geometric transformations (analytic equations): Translations, symmetries, rotations, orthogonal transformations, isometries. Homotheties and inversions. 2. Circle, ellipse, hyperbola, parabola: Common definition of conics. Optical properties. Reduction of conics to canonical form. 3. Vectors in space: Vector product. Cartesian frame. Coordinate system in space. Line and plane in space. Analytic equations. 4. Distance from a point to a plane. Volume of a tetrahedron. Geometric transformations in space. 5. Sphere, ellipsoid, hyperboloids, paraboloids: Reduction of quadrics to canonical form. Intersection of the cone with planes.
10.	DS	**Object Oriented Programming I**	**Object oriented programming I** 1. Basic elements of object-oriented programming: classes, objects, variables, methods 2. The inheritance mechanism in object-oriented languages 3. Implementing the polymorphism property in Java 4. Abstract classes 5. Interfaces. Multiple inheritance 6. Building graphical user interfaces in Java 7. Event management in Java 8. Managing object collections in Java 9. Reading data in Java 10. Writing data in Java
Anul II			
Nr. crt.	**Categoria formativă**	**Plan învățământ site**	**Subiecte**

1.	DF	**Graphs Algorithms**	**GRAPHS ALGORITHMS** 1. Algorithms for determining strongly connected components (description of an algorithm). 2. Algorithms for determining minimum paths (description of an algorithm). 3. Maximum flow in networks. Ford-Fulkerson algorithm. 4. Trees. Equivalent definitions. Prim algorithm (description) 5. Tree traversal. Infix, prefix, postfix forms. Partial minimum cost tree (description of an algorithm).
2.	DF	**Operating Systems**	**Operating systems** 1. Processes: process definition, process creation, the process image/memory layout, user vs kernel split. 2. Process states and transitions: 5-state and 7-state transition diagram, process exit statuses, the Process Control Block (PCB). 3. Process hierarchies in Linux: system calls for forking and process image replacement, orphan and zombie processes 4. Process and CPU isolation and protection: isolation mechanisms, context switches, handling system calls. 5. Interprocess communication mechanisms: signals, pipes, message queues, shared memory. 6. Thread synchronization using mutex locks: variants, mutex lock-based solutions for synchronization problems. 7. Thread synchronization using semaphores: variants, semaphore-based solutions for synchronization problems. 8. CPU scheduling algorithms: enumeration, description, the concept of preemption, example application of the algorithms on a process queue, waiting and turnaround times.

			9. Virtual memory management: page replacement algorithms, algorithm comparison on a reference string, Bélády's anomaly. 10. The buddy system kernel memory allocation algorithm: flowchart/pseudocode, example, buddy address calculation
3.	DF	**Formal Languages and Compilers**	**Formal Languages and Compilers** 1. Demonstrate the equivalence between non-deterministic and deterministic finite automata. 2. State and prove the pumping lemma for regular languages. 3. Prove that the language L = {ww | w ∈ {a, b}*} is not a regular language. 4. State and prove the pumping lemma for context-free languages and show that the language L = {$a^{2\hat{}n}$ | n ≥ 0 } is not context-free. 5. Consider the context-free grammar G: S → AB S → BC A → BA A → a B → CC B → b C → AB C → a and the string w = baaba. Show using the CYK algorithm that w∈L(G). Construct a derivation tree for w.

			6. Algorithmically construct the regular expression equivalent to the automaton M in the figure:  7. Construct a finite automaton M that accepts the set of all words that represent signed rational numbers with a finite number of decimals (e.g., +12.36, -0.98, 15 are words in the accepted language). Show that +12.36 belongs to the language accepted by M. 8. Define the concept of a regular expression and construct the regular expression that indicates the language $L=\{w^n c w^n \mid w \in \{a, b\}^* \text{ and } n<4\}$. Construct a Java method that checks if a string belongs to the language L. 9. Construct a pushdown automaton M that accepts the set of all words that represent correctly placed parenthesis strings (where $\Sigma = \{(,), [,]\}$) and show that w = [()]() belongs to the language accepted by M. 10. State the minimization algorithm for a finite automaton (including the lemmas used) and construct the equivalent minimal automaton for the automaton M given in the figure:

			
4.	DF	**Mathematical and Computational Logic**	**Mathematical and Computational Logic** 1. Conjunctive normal form of a formula in propositional logic. 2. Disjunctive normal form of a formula in propositional logic. 3. An algorithm for minimizing conjunctive normal form. 4. Skolem normal form. Example. Normalization algorithm. 5. Predicate calculus: alphabet, terms, formulas, axioms, deduction rules. Examples.
5.	DF	**Computational Geometry**	**Computational geometry** 1. 3SUM algorithm 2. Localization in the partition induced by a triangle 3. Localization in the partition induced by a simple polygon 4. Localization in the partition induced by a convex polygon 5. Localization in the partition induced by a PSLG – slabs method 6. Localization in the partition induced by a PSLG – chains method 7. Limitation problems 8. 2D tree 9. Convex Hull – Graham algorithm 10. Convex Hull – Jarvis algorithm

6.	DC	**Differential and Partial Differential Equations**	**Differential and Partial Differential Equations** 1. Separable Differential Equations. 2. Bernoulli Differential Equations. 3. Lagrange and Clairaut Differential Equations. 4. Exact Differential Equations. Integrating Factors. 5. Linear n^{th} Order Differential Equations with Constant Coefficients.
7.	DS	**Probability Theory and Elements of Mathematical Statistics**	**Probability Theory and Elements of Mathematical Statistics** 1. Probability: definitions, properties, types of probability 2. Conditional probability: definition, conditional multiplication formula, example 3. Conditional probability: definition, formula of total probability, Bayes formula, example 4. Independence of random events: definition, properties, example 5. Classical probability schemes: Poisson's scheme, examples 6. Expected value of a random variable: definition, properties, example 7. Variance of a random variable: definition, properties, example 8. Histogram: definition, types, examples 9. Location parameters in descriptive statistics: definitions, examples 10. Spread parameters in descriptive statistics: definitions, examples
8.	DS	**Numerical Analysis**	**Numerical Analysis** 1. Bernstein Approximation Polynomials (definition, properties, examples) 2. Lagrange Interpolation Polynomials (definition, properties, examples) 3. Quadrature Formulas. Trapezoidal Rule (definition, properties, composite trapezoidal rule, examples) 4. Quadrature Formulas. Simpson Rule (definition, properties, composite Simpson rule, examples)

			5. Pivoting Techniques for Direct Methods of Solving Square Linear Systems (definition, how the Gaussian Elimination algorithm is modified and its time complexity, examples)
9.	DF	**Databases**	**Databases** 1. Entity Relationship Data Model (classical elements, extensions, modelling traps; examples). 2. Primitive constraints in the Relational Data Model (definition for the 5 types of primitive constraints; the algorithm for the calculation of keys of a relation; practical examples). 3. Translation of ERD schemes into RDM (algorithm; types of anomalies; complex examples). 4. Normal forms (1NF, 2NF, 3NF; types of anomalies; examples). 5. SQL SELECT statement: SELECT, FROM clauses, JOIN OPERATIONS (syntax, semantics, examples). 6. SQL SELECT statement: WHERE clause; NULL values, SQL String operations, BETWEEN…AND operator, IN/NOT IN operator (syntax, semantics, examples). 7. SQL SELECT statement: aggregation functions; GROUP BY, HAVING clauses; examples. 8. Subqueries in the SQL SELECT statement (subqueries in SELECT, FROM, WHERE, HAVING; ANY/ALL/SOME keywords; IN/NOT IN, EXISTS/NOT EXISTS; examples). 9. SQL Data Definition Language statements (CREATE TABLE, ALTER TABLE, DROP TABLE; examples). 10. SQL Data Manipulation Language statements (INSERT, UPDATE, DELETE; subqueries in the WHERE clause; examples).
10.	DS	**Web Technologies**	**Web technologies**

			1. Essential elements of HTML 2. Superset of CSS 3. Grid in CSS 4. Flexbox in CSS 5. Javascript Object Notation versus XML 6. Javascript objects 7. Javascript functions 8. Storage in browser 9. Extensions in Javascript 10. AJAX technology
11.	DS	**Object Oriented Programming II**	**Object oriented programming II** 1. Swing tables management 2. Swing trees management 3. Functional interfaces. Using functional interfaces in writing element flows 4. Programming threads in Java 5. Programming databases in Java
Anul III			
Nr. Crt.	**Categoria formativă**	**Plan invățământ site**	**Subiecte**
1.	DS	**Databases Management Systems**	**Databases Management Systems** 1. Concurrency management in a relational DBMS: transactions (definition, ACID properties), conflicting operations (definition), conflict-serializable schedules (definition), locks (definition, classification), 2PL and Strict-2PL protocols (definition, examples)

			2. Query optimization in a relational DBMS: Join operator implementation techniques (features, cost, examples), general indexing rules in relation to the JOIN operator 3. Data recovery in a relational DBMS: failure types, recovery strategies, UNDO/REDO operations, rules for writing the transaction log, the WARM RESTART algorithm (steps, examples). 4. Methods of accessing records (classification, examples) and the main indexing structures (classification, characteristics) in a relational DBMS 5. Query optimization in a relational DBMS: general indexing rules, index function, IOT, indexing in relation to WHERE and ORDER BY clauses, DML statements (characteristics and examples).
2.	DF	**Computer Networks**	**Computer Networks** 1. OSI and TCP/IP Reference Models 2. TCP/IP Protocol Suite 3. Data Link Layer Switching 4. IPv4 and IPv6 Logical Addressing 5. IP Address Sub-Allocation (Subnetting) 6. VLSM (Variable Length Subnet Mask) Method 7. Static Routing in IP Networks 8. Dynamic Routing based on Distance Vectors – RIP Protocol 9. Network Communication through Sockets (Client-Server TCP, Client-Server UDP - Python Applications) 10. HTTPS Protocol: Authenticating a Site Through a Certificate
3.	DS	**Advanced Programming Techniques**	1. Define the concept of Binding Source at the level of a connection with a database. Provide a relevant example. 2. Define how the connection to the database is established. Example: C# connection to an MS Access database. 3. Describe how you can access/modify multimedia elements in an MS Access database. Example: accessing an image from a database field.

			4. Define a database structure using multimedia-type fields. Relevant example describing the properties of an image-type field. 5. Describe how visual controls can be defined in MS Access. Example: button control and setting visibility in the application. 6. Define the association table. Create relationships between many-to-many tables. Example: using an association table. 7. Accessing tables from a password-protected database. 8. Handling errors in the programming language that accesses the database. Example. 9. Define the concept of validation through referential integrity in a database. Example using integrity access from the C# programming language. 10. Define the concept of constraint in a database. C# example that applies the constraint on table records.
4.	DF	**Artifficial Intelligence**	**Artificial Intelligence** 1. Intelligent Agents (Agent Definition, PEAS Description, Agent Types) 2. State-Based Models for Search Problems (State, Search Space, Transitions Between States, Search Algorithm; Practical Examples). 3. Uninformed Search Methods (Depth-First Search (DFS), Breadth-First Search (BFS), Practical Examples). 4. Informed Search Methods (Greedy Best First Search, A* Search; Practical Examples). 5. Constraint Satisfaction Problems (Problem Formulation, Backtracking for CSP, Heuristics for CSP (MRV, MCV, LCV); Examples) 6. Heuristic Local Search Algorithms (Hillclimbing, Simulated Annealing; Practical Examples). 7. Genetic Algorithms (Terminology, General Algorithm Schema, Selection Schemes).

			8. Genetic Algorithms (Solution Encoding/Representation, Fitness Function; Practical Examples). 9. Genetic algorithms (Genetic operators (mutation/crossover) for different types of solution representations; practical examples). 10. Particle Swarm Optimization metaheuristics (terminology, algorithm elements, pseudocode; practical example).
5.	DS	**Optimization Techniques**	**Optimization Techniques** 1. Simple and multiple, univariate and bivariate regression 2. TF-IDF scheme for search engines 3. Activation functions for artificial neural networks (ANNs) and forward propagation mechanism 4. Backpropagation mechanism in artificial neural networks 5. Artificial neural networks (ANNs) for modeling logic operators NOT, AND, OR, and XNOR
6.	DS	**Computer Graphics**	**Computer graphics** 1. Geometric transformations - translation / scaling / rotation around an Ox axis (Oy or Oz) 2. Color representation systems - RGB / CMY / HSV / HLS 3. Shading models - constant / Gouraud / Phong 4. Lighting methods - ambient / diffuse reflection / specular reflection 5. Animation techniques - key-frames / interpolation
7.	DS	**Web Application Development**	**Web application development** 1. Rounting in web applications 2. MVC architectures in web applications development 3. Object-Relational Mapping in ASP.NET Core 4. Design patterns in web applications development

			5. Model binding in ASP.NET Core 6. Containerization of web applications 7. Middleware in ASP.NET Core 8. Securing web applications 9. HTTP transactions 10. Dynamic forms in Asp.Net Core
8.	DS	**Scientific Computing Algorithms (Pattern Recognition)**	**Scientific Computing Algorithms (Pattern Recognition)** 1. NN and k-NN algorithms for face recognition 2. Eigenfaces algorithm for face recognition 3. Tensors. Data manipulation (folding and unfolding tensors) 4. k-means clustering algorithm (2 variants) 5. Convolutional neural networks (CNNs): components, various architectures, forward propagation 6. Performance metrics for classification problems in supervised learning
9.	DS	**Design and Programming Environments**	1. Define Resources, events, agents in the context of REA modelling for bussiness processes. 2. Define the REA extension the type concept. Exemplify. 3. Define the cardinalities for association relations. Exemplify 4. Define the REA extension - detail event concept. Exemplify through examples. 5. Define the way to specify attributes of the concepts in the REA model. 6. Define the database design using tables and attributes. 7. The design and implementation of the concepts in the REA model
10.	DS	**Networks Administration**	1. Network attacks. 2. Mitigation of network attacks. 3. The Spanning Tree Protocol (STP) algorithm. 4. The concept of Link Aggregation. 5. The DHCP protocol and its associated DORA process.