



UNIVERSITATEA DIN CRAIOVA

Facultatea de Automatică, Calculatoare și Electronică

**Departamentul de Calculatoare și Tehnologia Informației**

## **Examen de diplomă, promoția 2025**

### **TEMATICA propusă pentru Proba I**

#### **Evaluarea cunoștințelor fundamentale și de specialitate**

#### **Programul de studiu: “Calculatoare în limba engleză”**

#### **Domeniul: “ Calculatoare și Tehnologia Informației”**

##### **1. Computer Programming**

1. Programming Languages and Programming Paradigms
2. Algorithms: Design and Implementation
3. Data Structures
4. Expressing the Algorithms
5. Algorithm Analysis: Theoretical and Practical Validation
6. Programming in C: Program Structure and Functions. Preprocessing. Basic I/O Operations
7. Programming in C: Data and Control Structures
8. Programming in C: Arrays and Pointers
9. Programming in C: File I/O. The Standard Library

##### **2. Logical Design**

1. Boolean Algebra: Axioms and Theorems
2. Switching functions: Switching functions used in computing technique (constants, NOT, AND, OR, NAND, NOR, XOR, XNOR), Relations between switching functions, Properties of switching functions NAND, NOR, XOR
3. Logic forms: Minterms and maxterms, Canonical representations of switching functions (CDNF, CCNF), Normal forms (DNF, CNF), Representation of switching functions in NAND or NOR logic
4. Minimization of switching functions using KV and VEM maps
5. Combinational logic circuits in MSI: Logical Multiplexors (MUX), Logical Decoders (DEC), Binary adders (ADDER), Binary Comparators (COMP)
6. Sequential Logic Circuits: Mealy and Moore model
7. Flip-flops (RS, JK, D, T)
8. Sequential logic circuits in MSI (Registers, Counters)

##### **3. Computer Programming - Programming Techniques**

1. Algorithm complexity analysis
2. Algorithm correctness
3. Sorting algorithms
4. Abstract data types
5. Lists, stacks and queues
6. Graphs and trees. Concepts, representation and traversal
7. Divide and conquer algorithms
8. Dynamic programming
9. Greedy algorithms
10. Backtracking algorithms

#### **4. Data Structures and Algorithms**

1. Binary search trees
2. Height balanced trees
3. B trees
4. Graphs - Minimum cost spanning trees

#### **5. Object Oriented Programming**

1. Classes
  - Method overloading. Definition. Example.
  - Public/protected/private Definition. Example.
  - Constructor. Definition. Example.
  - Static member of a class. Definition. Example.
  - The keyword "this" Definition. Example.
  - Class composition. Definition. Example.
2. Inheritance
  - Definition. Advantages
  - Public/protected/private related to inheritance. Explanations. Example.
  - What is method overriding ?
3. Template classes
  - Definition. Use case. Example.
4. Abstract classes. Definition. Where to use it. Example
5. Interfaces. Definition. Where to use it. Example
6. Virtual. Definition. Where to use it. Example

#### **6. Computer Systems Architecture**

1. The von Neumann's model of a digital computer
2. Instruction cycle
3. General structure of a CPU
4. EEC – Mode of operation: Fetch phase and Execute phase
5. Memory addressing techniques: based addressing, paged addressing, indexed addressing

#### **7. Operating Systems**

1. Process Management
2. Synchronization in Linux

### 3. Synchronization in Windows

## **8. Databases**

1. Entity-Relationship Model
2. Relational Model
3. Basic SQL

## **9. Computer Networks**

1. IP addresses, network masks
2. TCP/IP Four Layers Architecture Model
3. ARP
4. DHCP
5. DNS

## **10. Software Engineering**

1. Agile Software Engineering.
2. Software Architecture
3. Reliable programming
4. Testing
5. Code Management