

## Programarea calculatoarelor și limbaje de programare

## Computer programming and programming languages

### Obiectiv principal

Introducerea, înțelegerea și aprofundarea noțiunilor fundamentale privind structurile de control fundamentale și programarea calculatoarelor.

### Course Objective

Introduction, understanding and deepening of the fundamental notions regarding fundamental control structures and computer programming.

### Curs

#### 2 ore/săptămână, total 28 ore

- Limbajul de programare C++: Introducere. Structura unui program în limbajul C++. Exemplu de program în C++.
- Tipuri fundamentale de date: Tipuri fundamentale. Manipulatori de tip. Declararea variabilelor. Constante.
- Tipuri fundamentale de date: Expresii aritmetice. Tablouri. Instrucțiuni de atribuire. Instrucțiuni intrare – ieșire.
- Manipulatori. Funcții utilizate pentru citirea sau scrierea datelor. Operatorul sizeof. Operatorii ++ și –.
- Operații cu numere întregi la nivel de bit. Structuri de control fundamentale: operatorul if, Operatorul switch.
- Structuri de control fundamentale: Operatorul ?. Operatorul while. Operatorul do-while. Operatorul for.
- Instrucțiunile Break și Continue. Funcții: Funcții predefinite. Funcții definite de programator.
- Funcții: Compilarea separată a funcțiilor, Funcții cu argumente tablouri. Supraîncărcarea funcțiilor. Funcții recursive. Funcții generice. Transmiterea parametrilor către funcții.
- Pointeri. Pointeri și tablouri. Referințe. Pointeri la funcții.
- Structuri și uniuni tip C: Structuri și uniuni. Aplicații : funcții de timp.
- Clase: Conceptele programării cu obiecte. Definiția unei clase. Pointerul this. Spați de nume. Constructori și destructori. Funcții prietene. Fișiere standard în C++. Șiruri standard tip C++.
- Moștenirea: Pointeri la obiecte. Moștenirea. Tipuri de acces. Funcții virtuale. Polimorfism. Date și funcții statice.
- Fișiere tip C++: Fișiere tip text. Fișiere binare.
- Tratarea excepțiilor: Excepții. Excepții lansate de funcții. Excepții standard.

### Laborator

#### 2 ore pe săptămână, total 28 ore

- Reguli de protecția muncii. Prezentarea lucrărilor de laborator.
- Mediul de programare C++. Etapele generării unui proiect. Exemple.
- Operații aritmetice. Exemple.
- Structuri de control fundamentale: Instrucțiunea if. Operatorii ++ și --.
- Structuri de control fundamentale: Instrucțiunile while, do-while și for. Operatori de deplasare. Operatori logici pe biți.
- Structuri de control fundamentale: Utilizarea vectorilor și matricelor.
- Utilizarea funcțiilor.
- Operații cu șiruri de caractere tip C++.
- Pointeri și referințe. Alocarea dinamică a memoriei.
- Prelucrarea fișierelor în C++. Structuri și uniuni. Clase.
- Moștenirea. Funcții virtuale. Polimorfism. Biblioteca de șabloane standard.

### Course

#### 2 hours weekly, 28 hours total

- C ++ programming language: Introduction. Structure of a program in C ++. Example program in C ++.
- Fundamental types of data: Fundamental types. Type manipulators. Declaration of variables. Constants.
- Fundamental types of data: Arithmetic expressions. Pictures. Awarding instructions. Input - output instructions.
- Manipulators. Functions used for reading or writing data. The sizeof operator. ++ and – operators.
- Operations with integers at bit level. Fundamental control structures: the if operator, the Switch operator.
- Fundamental control structures: The operator? The operator while. The do-while operator. The operator is.
- Break and Continue instructions. Functions: Predefined functions. Functions defined by the programmer.
- Functions: Separate compilation of functions, Functions with array arguments. Function overloading. Recursive functions. Generic functions. Transmission of parameters to functions.
- Pointers. Pointers and paintings. References. Pointers to functions.
- Structures and unions type C: Structures and unions. Applications: time functions.
- Classes: Object programming concepts. Defining a class. This pointer. Namespace. Builders and destroyers. Friendly functions. Standard C ++ files. Standard C ++ rows.
- Inheritance: Pointers to objects. Inheritance. Types of access. Virtual functions. Polymorphism. Static data and functions.
- C ++ files: Text files. Binary files.
- Handling exceptions: Exceptions. Exceptions thrown by functions. Standard exceptions.

### Laboratory

#### 2 hours weekly, total 28 hours

- Rules of labor protection. Presentation of laboratory work.
- C ++ programming environment. Stages of project generation. Examples.
- Arithmetic operations. Examples.
- Fundamental control structures: the if statement. ++ and - operators.
- Fundamental control structures: while, do-while and for instructions. Moving operators. Logical bit operators.
- Fundamental control structures: Use of vectors and matrices.
- Use of functions.
- Operations with C ++ character strings.
- Pointers and references. Dynamic memory allocation.
- C ++ file processing. Structures and unions. Classes.
- Inheritance. Virtual functions. Polymorphism. Standard template library.