



ROMÂNIA
MINISTERUL EDUCAȚIEI ȘI CERCETĂRII
UNIVERSITATEA DIN CRAIOVA
FACULTATEA DE INGINERIE ELECTRICĂ

Craiova, Bdul. Decebal, nr. 107, 200440,
tel/fax: +40-251-436447, www.ie.ucv.ro
e-mail: secretariat@ie.ucv.ro



NUMELE DISCIPLINEI	COURSE NAME
DATE GENERALE	GENERAL INFORMATION
Program de studii: Inginerie Electrica si Calculatoare Anul: I Semestrul: 2 Credite ECTS: 5 Tipuri de activități: Curs, Laborator Tip evaluare: E Titular: Ș.I. dr. ing. Andreea-Maria NEACĂ	Study Program: Electrical Engineering and Computers Year: I Semester: 2 ECTS Credits: 5 Types of activities: Lecture, Laboratory Assessment: E Lecturer: Ș.I. dr. ing. Andreea-Maria NEACĂ
SCOPUL DISCIPLINEI	COURSE PURPOSE
Această disciplină este fundamentală pentru domeniul de studii. Aici , studentii învață să utilizeze mediul Visual C++, pentru a dezvolta algoritmi de programare, concepte, principii și metode de bază din matematică, fizică si inginerie.	This discipline is fundamental to the field of study. Here, students learn to use the Visual C++ environment, to develop programming algorithms, concepts, principles, and basic methods in mathematics, physics, and engineering.
CERINȚE PREALABILE	PREREQUISITES
Studentul trebuie să posede cunoștințe de specialitate dobândite la următoarele discipline: Analiză matematică, Algebră liniară și geometrie diferențială, Informatică aplicată, Comunicare.	Students must have specialized knowledge acquired in the following subjects: Mathematical Analysis, Linear Algebra and Differential Geometry, Applied Computer Science, Communication.
OBIECTIVE	OBJECTIVES
• Înțelegerea problemelor de dimensiuni reduse expuse în limbaj natural și dezvoltarea unor soluții sub forma programelor de calculator; • Înțelegerea codului sursă scris de alți programatori și abilitatea de a analiza critic acel cod; • Proiectarea și implementarea programelor în limbajul C folosind o abordare structurată / modulară; • Învățarea unui stil de programare adecvat; • Identificarea erorilor de programare, detectarea cauzelor și corectarea acestora.	• Understanding small-scale problems expressed in natural language and developing solutions in the form of computer programs; • Understanding source code written by other programmers and the ability to critically analyze that code; • Designing and implementing programs in the C language using a structured/modular approach; • Learning an appropriate programming style; • Identifying programming errors, detecting their causes, and correcting them.
CONȚINUT CURS	LECTURE CONTENT
1. Limbajul de programare C++: Introducere. Structura unui program în limbajul C++. Exemplu de program în C++. 2. Tipuri fundamentale de date: Tipuri fundamentale. Manipulatori de tip. Declararea variabilelor. Constante. 3. Tipuri fundamentale de date: Expresii aritmetice. Tablouri. Instrucțiuni de atribuire. Instrucțiuni intrare – ieșire. 4. Manipulatori. Funcții utilizate pentru citirea	1. C++ programming language: Introduction. Structure of a program in C++. Example of a program in C++. 2. Fundamental data types: Fundamental types. Type manipulators. Declaring variables. Constants. 3. Fundamental data types: Arithmetic expressions. Arrays. Assignment statements. Input-output statements. 4. Manipulators. Functions used for reading or writing data. The sizeof operator. The ++ and – operators. 5. Bitwise integer operations. Fundamental control



ROMÂNIA
MINISTERUL EDUCAȚIEI ȘI CERCETĂRII
UNIVERSITATEA DIN CRAIOVA
FACULTATEA DE INGINERIE ELECTRICĂ

Craiova, Bdul. Decebal, nr. 107, 200440,
tel/fax: +40-251-436447, www.ie.ucv.ro
e-mail: secretariat@ie.ucv.ro



sau scrierea datelor. Operatorul sizeof. Operatorii ++ și – 5. Operații cu numere întregi la nivel de bit. Structuri de control fundamentale: operatorul if, Operatorul switch. 6. Structuri de control fundamentale: Operatorul ?. Operatorul while. Operatorul do-while. Operatorul for. 7. Instrucțiunile Break și Continue. Funcții: Funcții predefinite. Funcții definite de programator. 8. Funcții: Compilarea separată a funcțiilor, Funcții cu argumente tablouri. Supraîncărcarea funcțiilor. Funcții recursive. Funcții generice. Transmiterea parametrilor către funcții. 9. Pointeri. Pointeri și tablouri. Referințe. Pointeri la funcții. 10. Structuri și uniuni tip C: Structuri și uniuni. Aplicații : funcții de timp. 11. Clase: Conceptele programării cu obiecte. Definirea unei clase. Pointerul this. Spații de nume. Constructori și destructori. Funcții prietene. Fișiere standard în C++. Șiruri standard tip C++. 12. Moștenirea: Pointeri la obiecte. Moștenirea. Tipuri de acces. Funcții virtuale. Polimorfism. Date și funcții statice. 13. Fișiere tip C++: Fișiere tip text. Fișiere binare. 14. Tratarea excepțiilor: Excepții. Excepții lansate de funcții. Excepții standard.	structures: "if" operator, "switch" operator. 6. Fundamental control structures: "?" operator. "while" operator. "do-while" operator. "for" operator. 7. Break and Continue statements. Functions: Predefined functions. Functions defined by the programmer. 8. Functions: Separate compilation of functions, Functions with array arguments. Function overloading. Recursive functions. Generic functions. Passing parameters to functions. 9. Pointers. Pointers and arrays. References. Pointers to functions. 10. C-type structures and unions: Structures and unions. Applications: time functions. 11. Classes: Concepts of object-oriented programming. Defining a class. "This" pointer. Namespaces. Constructors and destructors. Friend functions. Standard files in C++. Standard C++ strings. 12. Inheritance: Pointers to objects. Inheritance. Access types. Virtual functions. Polymorphism. Static data and functions. 13. C++ files: Text files. Binary files. 14. Handling exceptions: Exceptions. Exceptions thrown by functions. Standard exceptions.
METODE DE EVALUARE	EVALUATION METHODS
Examen, Evaluare continuă	Exam, Continuous Assessment
COMPETENȚE DOBÂNDITE	ACQUIRED COMPETENCIES
• Studentul învață să: aplice criterii și metode de evaluare pentru identificarea, modelarea, experimentarea, analiza și aprecierea calitativă și cantitativă a fenomenelor și proceselor specifice domeniului fundamental folosind inclusiv tehnologii digitale. • opereze cu concepte, principii și metode de bază din matematică, fizică, chimie, desen tehnic, economie și informatică. • practice raționamentul logic, evaluarea și autoevaluare în luarea deciziilor.	• Students learn to: apply evaluation criteria and methods for identifying, modeling, experimenting, analyzing, and qualitatively and quantitatively assessing phenomena and processes specific to the fundamental field, including the use of digital technologies. • operate with basic concepts, principles, and methods in mathematics, physics, chemistry, technical drawing, economics, and computer science. • practice logical reasoning, evaluation, and self-evaluation in decision-making.
Contact: [andreea.neaca@edu.ucv.ro]	Contact: [andreea.neaca@edu.ucv.ro]
Ultima actualizare: [01.10.2025]	Last update: [01.10.2025]