COURSE SYLLABUS

University	UNIVERSITY OF ORADEA
Faculty	FACULTY OF ENERGY ENGINEERING AND INDUSTRIAL MANAGEMENT
Study program*	KNITTING AND CLOTHING TECHNOLOGY

I. Course Name: COMPUTERS PROGRAMMING AND PROGRAMMING LANGUAGES II

II. Course Details

No of hours/week						
Code	Semester	Credits	Lecture	Seminar	Laboratory	Project
IEMI-0163	2	5	1	-	3	-

III. Course coordinator (title, name, surname, e-mail):
Lecturer Vasile Moldovan, moldovan@uoradea.ro

IV. Course objectives

Understand the fundamentals of C++ programming, including program structure, data types, operators, and control statements. Apply core programming techniques, such as arrays, pointers, dynamic memory, and functions (including recursion and overloading). Implement object-oriented programming principles, including classes, objects, constructors, destructors, and advanced features like inheritance and friend functions.
Develop practical problem-solving skills through hands-on exercises in the Borland C++ environment, integrating theoretical concepts into real applications.

V. Course content	**No. of hours**
V.1. Lecture (chapters/subchapters and paragraphs)	
C1. Basic elements of the language. Structure of a C++ program.	1
C2. Input/Output devices. Console input/output functions.	1
C3. Predefined data types, constants, variable declarations.	1
C4. Operators and expressions (operators, operator precedence and associativity, type conversions in expressions).	1
C5. Statements (expression statements, compound statements, decision statements, looping statements, jump statements).	1
C6. Arrays and character strings (one-dimensional arrays, character strings, multidimensional arrays).	1
C7. Pointer variables (assignment operations with pointers, arithmetic operations with pointers, arrays and pointers, dynamic variables).	1
C8. Functions (prototypes, defining functions, parameter passing, function return values, function pointers, recursive functions, default parameter values, function overloading, inline functions).	1
C9. User-defined data types (enumerations, structures, bit fields).	1
C10. Classes and objects (the class type, self-reference, constructors and destructors, static members, pointers to class members).	1
C11. Using objects.	1
C12. Friend functions and friend classes.	1
C13. User-defined type conversions.	1
C14. Inheritance. Derived classes.	1
V.2. Laboratory	
L1. Introduction to the Borland C++ programming environment. Structure of a C++ program.	3
L2. Input/Output devices. Console input/output functions.	3
L3. Predefined data types, constants, variable declarations.	3
L4. Operators and expressions – operator precedence and associativity, type conversions in expressions.	3
L5. Expression statements; compound statements; decision statements.	3
L6. Looping statements; jump statements.	3
L7. One-dimensional arrays and character strings.	3

L8. Multidimensional arrays.	3
L9. Pointer variables; operations with pointers.	3
L10. Arrays and pointers; dynamic variables.	3
L11. Functions – parameter passing, function return values.	3
L12. Classes and objects (the class type, self-reference, constructors and destructors).	3
L13. Friend functions and friend classes.	3
L14. Inheritance. Derived classes. Class hierarchies. Multiple inheritance.	3

VII. Grading criteria

Activities	Assessment	% of final grade
Exam	1 final verification test - Written exam (or multiple-choice test) on theoretical knowledge, including practical application. Evaluation Criteria: Correctness and completeness of knowledge; Logical coherence; Degree of assimilation of specialized terminology.	70 %
Laboratory	Ability to apply knowledge in practice; Criteria regarding attitudinal aspects: conscientiousness, interest in individual study. Periodic written assessments. Active participation in laboratory sessions.	20 % 10 %

VIII. Learning outcomes:

- Write and debug programs using fundamental language constructs, data types, operators, and control structures.
- Develop modular programs by applying arrays, pointers, dynamic memory allocation, and functions.
- Design and implement object-oriented solutions, utilizing classes, objects, constructors.
- Apply programming concepts in practical scenarios in the Borland C++ environment.

Course coordinator,
Lecturer Vasile MOLDOVAN, PhD., Eng.