

SYLLABUS

1. Program information

1.1 Higher education institution	Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Electronics, Telecommunications and Information Technology
1.3 Department	Bases of Electronics
1.4 Field of study	Electronic engineering, telecommunications and information technologies
1.5 Cycle of study	Bachelor of Science
1.6 Study program / Qualification	Electronic engineering, telecommunications and information technologies/ Engineer
1.7 Form of education	IF – full-time education

2. Course information

2.1 Course title	Computer Programming and Programming Languages II	Course code	11.00
2.2 Course coordinator	Professor, PhD Eng. Mircea-Florin Vaida, Mircea.Vaida@com.utcluj.ro		
2.3 Seminar/ laboratory/ project/ practical activity coordinator	Professor, PhD Eng. Mircea-Florin Vaida, Mircea.Vaida@com.utcluj.ro		
2.4 Year of study	I	2.5 Semester	2
		2.6 Type of assessment	V
2.7 Course status	Formative category		DF
	Optionality		DOB

3. Estimated total time

3.1 Hours per week	4	of which:	3.2 Lecture	2	3.3 Seminar	0	3.3 Laboratory	2	3.3 Project	0	3.3 Practical training	0
3.4 Hours per semester	56	of which:	3.5 Lecture	28	3.6 Seminar	0	3.6 Laboratory	28	3.6 Project	0	3.3 Practical training	0
3.7 Time distribution (hours per semester) for individual study and assessment:												
(a) Assessment											3	
(b) Study using textbook, course materials, bibliography, and notes											34	
(c) Additional documentation in the library, on specialized electronic platforms, and in the field											9	
(d) Preparation for seminars/laboratories, assignments, reports, portfolios and essays											20	
(e) Tutoring											2	
(f) Other activities											1	
3.8 Total hours of individual study and assessment (sum of (3.7(a))...3.7(f))									69			
3.9 Total hours per semester (3.4+3.8)									125			

3.10 Number of credits	5.0
------------------------	-----

4. Prerequisites (where applicable)

4.1 curriculum	Basic knowledge from: - Computer programming – Languages 1
4.2 competencies	Basic knowledge of algorithms

5. Requirements (where applicable)

5.1. for the development of the lecture	Video-projector, screen, whiteboard
5.2. for the development of the seminar/ laboratory/ project	PCs with Internet access

6. Specific competencies acquired

Professional competencies	C3. Application of basic knowledge, concepts and methods related to computer-system architecture, microprocessors, microcontrollers, programming languages and techniques C3.1 Description of the operation of a computer system, of the basic principles of general-purpose microprocessor and microcontroller architecture, and of the general principles of structured programming. C3.2 Use of general-purpose programming languages and languages specific to microprocessor and microcontroller applications; explanation of the operation of automatic control systems using these architectures and interpretation of experimental results. C3.3 Solving concrete practical problems that include elements of data structures and algorithms, programming and the use of microprocessors or microcontrollers. C3.4 Development of programs in a general-purpose and/or specific programming language, from requirements specification to execution, debugging and interpretation of the results in relation to the processor used. C3.5 Implementation of projects involving hardware components (processors) and software (programming). C4. Design and use of low-complexity hardware and software applications specific to applied electronics C4.1 Definition of the concepts, principles and methods used in: computer programming, high-level and specific languages, CAD techniques for creating electronic modules, microcontrollers, computer-system architecture, programmable electronic systems, graphics and reconfigurable hardware architectures. C4.2 Explanation and interpretation of the requirements specific to hardware and software structures in the fields of: computer programming, high-level and specific languages, CAD techniques for creating electronic modules, microcontrollers, computer-system architecture, programmable electronic systems, graphics and reconfigurable hardware architectures. C4.3 Identification and optimization of hardware and software solutions for problems related to industrial electronics, medical electronics, automotive electronics, automation, robotics and consumer-goods production. C4.4 Use of appropriate performance criteria to evaluate, including by simulation, the hardware and software of dedicated systems or service activities using microcontrollers or low-/medium-complexity computer systems. C4.5 Design of dedicated equipment in the fields of applied electronics, using microcontrollers, programmable circuits or computer systems with a simple architecture, including the related programs. C5. Application of basic knowledge, concepts and methods from: power electronics, automatic systems, electrical energy management and electromagnetic compatibility C5.1 Definition of the specific elements that characterize electronic devices and circuits in the fields of: power electronics, automatic systems, electrical energy management, medical electronics, automotive electronics and consumer goods. C5.2 Qualitative and quantitative interpretation of the operation of circuits in the fields of: power electronics, automatic systems, electrical energy management, medical electronics, automotive electronics and consumer goods; analysis of operation from the standpoint of electromagnetic compatibility.
---------------------------	--

Professional competencies	C5.3 Preparation of technical specifications, installation and operation of equipment in the fields of applied electronics: power electronics, automatic systems, electrical energy management, medical electronics, automotive electronics and consumer goods. C5.4 Evaluation, based on technical quality and environmental impact criteria, of equipment in the fields of applied electronics: power electronics, automatic systems, electrical energy management, medical electronics, automotive electronics and consumer goods. C5.5 Design, using established principles and methods, of low-complexity subsystems in the fields of applied electronics: power electronics, automatic systems, electrical energy management, medical electronics, automotive electronics and consumer goods.
Transversal competence	

7. Expected learning outcomes

Knowledge	- The student/graduate identifies and describes basic concepts, principles and methods in mathematics, physics, chemistry, technical drawing, economy and computer science. - The student/graduate describes, identifies and summarizes elementary concepts and methods concerning computer-system architecture, microcontrollers, programming languages and techniques, as well as their application in concrete problems.
Skills	- The student/graduate operates with basic concepts, principles and methods in mathematics, physics, chemistry, technical drawing, economy and computer science. - The student/graduate performs engineering and economic calculations of medium complexity and associates them with graphical/alphanumeric representations or with representations specific to computer-aided design. - The student/graduate describes physico-chemical and economic phenomena and processes. - The student/graduate evaluates, based on specific performance criteria, the type of computing system, its architecture and the manner of use required for efficiently solving concrete problems. - The student/graduate specifies requirements, develops programs in general-purpose programming languages (C, etc.) and/or object-oriented languages (C++, Java, etc.), runs and debugs them, and interprets the results of the programs developed in order to solve a concrete problem.
Responsibility and autonomy	- The student/graduate applies the values of ethics and professional deontology specific to the engineering profession. - The student/graduate applies logical reasoning, evaluation and self-assessment in decision-making. - The student/graduate communicates effectively about engineering activities with a wide range of audiences.

Responsibility and autonomy	- The student/graduate is engaged in lifelong learning in order to acquire and implement knowledge, as needed, using appropriate learning strategies. - The student/graduate promotes dialogue, cooperation, respect for others and interculturality. - The student/graduate works effectively as a team member or as a team leader. - The student/graduate fluently uses at least one international language in technical communication. - The student/graduate behaves honorably, responsibly and ethically, in the spirit of the law, in order to ensure the reputation of the profession.
-----------------------------	---

8. Course objectives (based on the grid of specific competencies acquired)

8.1 General objective of the course	Development of competences in basic algorithms and C/C++. Object Oriented Programming (OOP) language in C++
8.2 Specific objectives	1. Theoretical knowledges about basic OOP in C++ language. 2. Practical abilities to use Visual Studio C++ IDE for OOP and algorithms applications.

9. Contents

9.1 Lecture	No. of hours	Teaching methods	Remarks
1. Recursive programming in C/C++. Stack management. Functional programming basic elements. Recursive and non-recursive programming methods. Backtracking.	2	Presentations, discussions	
2. Recursive and non-recursive programming methods. Variants of Backtracking method. Divide et impera method.	2		
3. Sorting and searching algorithms. Simple sorting: selection, insertion, interchange. Advanced sorting: merge sort, quick-sort.	2		
4. Introduction in Object Oriented Programming, OOP. Classes, Objects, members of a class, interfaces, 4 pillars of OOP.	2		
5. Constructors, destructors, methods calling in C++. Copy constructor, arrays of objects, visibility domain.	2		
6. Friend class and functions in C++. Static members. Struct and union in C++. Overloading methods.	2		
7. Overloading operators in C++.	2		
8. Inheritance in C++. Simple and multiple inheritances.	2		
9. Virtual classes and methods. Abstract classes.	2		
10. I/O operations in C++. iostream library, I/O with format, I/O state, manipulators functions	2		
11. ostream, istream si fstream classes. Overloading I/O operators. C++ files.	2		
12. Generic programming in C++.	2		
13. STL C++ library. C++ Exceptions.	2		

9.1 Lecture	No. of hours	Teaching methods	Remarks
14. Theoretical evaluation	2		
Bibliography - Vaida M., Bazele dezvoltarii aplicatiilor software in electronica si telecomunicatii, curs, litografia UTC-N, 1997 - Mircea-Florin Vaida, Petre G. Pop, Cosmin Strilețchi, Ligia Chiorean, Calin G. Login, Tehnologii avansate privind dezvoltarea aplicatiilor software in limbajul C/C++, Casa Cartii de Stiinta, 2006 - Ligia Chiorean, Mircea-Florin Vaida, Petre G. Pop, Cosmin Strilețchi, , Elemente de bază și obiectuale privind dezvoltarea aplicațiilor în limbajul de programare C/C++, UTPress, 2007/2008 - Mircea-Florin Vaida, Ligia-Domnica Chiorean, Lenuța Alboai, Petre Gavril Pop, Cosmin Strilețchi, Kuderna-Iulian Bența, Programarea în limbajul C/C++ cu elemente C++1y. Programare web C++, Casa Cartii de Stiinta, Cluj-Napoca, 2016 - Ligia-Domnica Chiorean, Kuderna-Iulian Bența, Mircea-Florin Vaida, Petre Gavril Pop, Cosmin Strilețchi, C/C++ - Ghid teoretic si practic, Casa Cartii de Stiinta, Cluj-Napoca, 2016 - Mircea-F. Vaida, Ligia D. Chiorean, Adriana Stan, Cosmin Strilețchi, Petre G. Pop, Ștefan-S. Dragoș - Aplicații de bază folosind C/C++. Elemente practice. Varianta bilingvă - Basic applications using C/C++. Practical elements. Bilingual variant, UTPRESS Cluj-Napoca, 2023 - Mircea-F. Vaida, Ligia D. Chiorean, Cosmin Strilețchi, Petre G. Pop, Ștefan-S. Dragoș - Algoritmi de bază și programarea orientată pe obiecte folosind C/C++. Elemente practice. Varianta bilingvă - Basic algorithms and object-oriented programming using C/C++. Practical elements. Bilingual version, UTPRESS Cluj-Napoca, 2024 ISBN 978-606-737-727-9, https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/727-9.pdf - course notes at https://helios.utcluj.ro/learn2code/index.php Supplementary materials: Other information: Assessment, supplementary support from: https://helios.utcluj.ro/learn2code/index.php			

9.2 Seminar /laboratory /project /practical training	No. of hours	Teaching methods	Remarks
1. Macro functions. Assertions. Inline functions. Functions with implicit parameters. Functions with a variable number of parameters. Overloading functions	2	Presentations, discussions	
2. Recursive functions. Functional programming.	2		
3. Recursive and non-recursive programming methods, backtracking, divide et impera: searching techniques.	2		
4. Sorting techniques.	2		
5. Classes, objects, class members.	2		
6. The access to a class's members	2		
7. Constructors. Destructors. Object arrays	2		
8. Friend functions and classes. Static members.	2		
9. Operators overloading.	2		
10. Simple and multiple inheritances	2		
11. Virtual methods and classes. Abstract classes.	2		
12. Input/output in C++. Overloading the I/O operators.	2		
13. Files in C++. Homework evaluation	2		
14. Final practical test and evaluation.	2		

9.2 Seminar /laboratory /project /practical training	No. of hours	Teaching methods	Remarks
Bibliography - Vaida M., Bazele dezvoltarii aplicatiilor software in electronica si telecomunicatii, curs, litografia UTC-N, 1997 - Mircea-Florin Vaida, Petre G. Pop, Cosmin Strilețchi, Ligia Chiorean, Calin G. Login, Tehnologii avansate privind dezvoltarea aplicatiilor software in limbajul C/C++, Casa Cartii de Stiinta, 2006 - Ligia Chiorean, Mircea-Florin Vaida, Petre G. Pop, Cosmin Strilețchi, , Elemente de bază și obiectuale privind dezvoltarea aplicațiilor în limbajul de programare C/C++, UTPress, 2007/2008 - Mircea-Florin Vaida, Ligia-Domnica Chiorean, Lenuța Alboaie, Petre Gavril Pop, Cosmin Strilețchi, Kuderna-Iulian Bența, Programarea în limbajul C/C++ cu elemente C++1y. Programare web C++, Casa Cartii de Stiinta, Cluj-Napoca, 2016 - Ligia-Domnica Chiorean, Kuderna-Iulian Bența, Mircea-Florin Vaida, Petre Gavril Pop, Cosmin Strilețchi, C/C++ - Ghid teoretic si practic, Casa Cartii de Stiinta, Cluj-Napoca, 2016 - Mircea-F. Vaida, Ligia D. Chiorean, Adriana Stan, Cosmin Strilețchi, Petre G. Pop, Ștefan-S. Dragoș - Aplicații de bază folosind C/C++. Elemente practice. Varianta bilingvă - Basic applications using C/C++. Practical elements. Bilingual variant, UTPRESS Cluj-Napoca, 2023 - Mircea-F. Vaida, Ligia D. Chiorean, Cosmin Strilețchi, Petre G. Pop, Ștefan-S. Dragoș - Algoritmi de bază și programarea orientată pe obiecte folosind C/C++. Elemente practice. Varianta bilingvă - Basic algorithms and object-oriented programming using C/C++. Practical elements. Bilingual version, UTPRESS Cluj-Napoca, 2024 ISBN 978-606-737-727-9, https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/727-9.pdf - laboratory materials available on the Helios platform,https://helios.utcluj.ro/learn2code/index.php Supplementary materials: Other information: Assessment, supplementary support from: https://helios.utcluj.ro/learn2code/index.php			

10. Alignment of the course contents with the expectations of the epistemic community, professional associations, and representative employers in the field related to the program

The course content and the teaching-learning-assessment methods are aligned with the current requirements of the Electronics, Telecommunications, and Information Technology Engineering field, considering the expectations of the academic and professional community, representative employers, as well as the national quality assurance standards in higher education. The course helps prepare students in line with employer expectations and the quality requirements set for engineering study programs.

11. Assessment

Activity type	11.1 Assessment criteria	11.2 Assessment methods (and type of assessment: continuous/ summative)	11.3 Weight in final grade
11.4 Lecture	Theoretical (T) written and oral test with questions/code	Written/oral test (T=33%) continuous	T = 33%
11.5 Seminar/ Laboratory/ Project /Practical activity	Solving a problem P on a computer (1 hour). Laboratory L will also be evaluated as activity and tests	Lab. evaluations and computer test (P=34%, L=33%), continuous	P+L = 67%
11.6 Minimum performance standard			

The final grade (N) is calculated as average of marks obtained in the evaluation of ongoing activities and application type: $N = (T + L + P) / 3.0$. The condition for obtaining the ECTS credits is that N and all components of the final grade to be higher than or equal to 5 (five).

Completion date:	Course coordinators	teaching rank, title First name LAST NAME	Signature
23.06.2026	Lecture	Professor, PhD Eng. Mircea-Florin Vaida	
	Applications	Professor, PhD Eng. Mircea-Florin Vaida	
		Assoc. Prof. PhD Eng. Cosmin Strilechi	

Approval of the Head of the Department coordinating the course, only if this differs from the department organizing the study program	Head of the Communications Department Prof. Virgil DOBROTĂ, Ph.D.
---	--

Date of approval by the Council of the Department of Bases of Electronics	Head of the Bases of Electronics Department Prof. Sorin HINTEA, Ph.D.
26.06.2026	
Date of approval by the Council of the Faculty of Electronics, Telecommunications and Information Technology	Dean, Prof. Ovidiu-Aurel POP, Ph.D.
30.06.2026	