Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti
Facultatea de Electronică, Telecomunicaţii şi
Tehnologia Informaţiei

COURSE DESCRIPTION

1. Program identification information

1.1 Higher education institution	**National University of Science and Technology Politehnica Bucharest**
1.2 Faculty	**Electronics, Telecommunications and Information Technology**
1.3 Department	**Applied Electronics and Information Engineering**
1.4 Domain of studies	Computers and Information Technology
1.5 Cycle of studies	Bachelor/Undergraduate
1.6 Programme of studies	Information Engineering

2. Date despre disciplină

2.1 Course name (ro) (en)				Programarea calculatoarelor şi limbaje de programare 1 Computer programming and programming languages 1			
2.2 Course Lecturer				Prof. Dr. Bogdan Emanuel IONESCU			
2.3 Instructor for practical activities				Prof. Dr. Bogdan Emanuel IONESCU			
2.4 Year of studies	1	2.5 Semester	I	2.6. Evaluation type	E	2.7 Course regime	Ob
2.8 Course type		F	2.9 Course code	04.F.01.O.004		2.10 Tipul de notare	Nota

3. Total estimated time (hours per semester for academic activities)

3.1 Number of hours per week	4	Out of which: 3.2 course	2.00	3.3 seminary/laboratory	2
3.4 Total hours in the curricula	56.00	Out of which: 3.5 course	28	3.6 seminary/laboratory	28
Distribution of time:					hours
Study according to the manual, course support, bibliography and hand notes Supplemental documentation (library, electronic access resources, in the field, etc) Preparation for practical activities, homework, essays, portfolios, etc.					38
Tutoring					0
Examinations					6
Other activities (if any):					0

3.7 Total hours of individual study	44.00
3.8 Total hours per semester	100
3.9 Number of ECTS credit points	4

4. Prerequisites (if applicable) (where applicable)

4.1 Curriculum	Not applicable.
4.2 Results of learning	Not applicable.

5. Necessary conditions for the optimal development of teaching activities (where applicable)

5.1 Course	Access to a video projection system and Internet. C/C++ code editing and compilation utility package.
5.2 Seminary/ Laboratory/Project	Individual access to PC systems equipped with C/C++ editors and compilers as well as an Internet connection.

6. General objective *(Reffering to the teachers' intentions for students and to what the students will be thought during the course. It offers an idea on the position of course in the scientific domain, as well as the role it has for the study programme. The course topics, the justification of including the course in the currcula of the study programme, etc. will be described in a general manner)*

Course: the discipline studies in its first part the general principles of structured programming and in the second part the fundamental problems of programming in C. The C language is presented in a gradual way, covering all the essential aspects of programming in C: data organization, instructions, arrays, functions, pointers.

Laboratory: mastering the fundamental aspects regarding the use of the C programming language. The applications include both the basic notions of the C language syntax and elements of program design. Problems with different degrees of difficulty are solved and the various possibilities of solving a particular problem are commented on, the way in which a program can be written more efficiently, as well as the typical errors that occur when developing a C program.

7. Competences *(Proven capacity to use knowledge, aptitudes and personal, social and/or methodological abilities in work or study situations and for personal and proffesional growth. They refflect the empolyers requirements.)*

Specific Competences	Application of fundamental knowledge, concepts and methods regarding computer system architecture, microcontrollers, programming languages and techniques.
Transversal (General) Competences	The ability to constantly inform and document oneself for personal and professional development by reading specialized literature.

8. Learning outcomes *(Synthetic descriptions for what a student will be capable of doing or showing at the completion of a course. The learning outcomes reflect the student's acomplishments and to a lesser extent the teachers' intentions. The learning outcomes inform the students of what is expected from them with respect to performance and to obtain the desired grades and ECTS points. They are defined in concise terms, using verbs similar to the examples below and indicate what will be required for evaluation. The learning outcomes will be formulated so that the correlation with the competences defined in section 7 is highlighted.)*

Knowledge	*The result of knowledge aquisition through learning. The knowledge represents the totality of facts, priciples, theories and practices for a given work or study field. They can be theoretical and/or factual.* • mastering the fundamental concepts of computer programming, • mastering programming in the C/C++ programming language, • mastering the knowledge to solve a basic programming problem, • mastering the knowledge to debug a program in the C/C++ language.

Skills	*The capacity to apply the knowledge and use the know-how for completing tasks and solving problems. The skills are described as being cognitive (requiring the use of logical, intuitive and creative thinking) or practical (implying manual dexterity and the use of methods, materials, tools and intrumentation).* • the ability to understand and explain a program written in C/C++ code, • the ability to design a simple algorithm, • the ability to validate the results of a C/C++ program, • the ability to identify programming solutions, • the ability to communicate and argue solutions.
Responsability and autonomy	*The student's capacity to autonomously and responsably apply their knowledge and skills.* • the ability to select and browse bibliographic sources, • the ability to promote and contribute with new solutions, • the ability to learn new concepts, • the ability to communicate information with other colleagues, • the development of autonomy in the learning process.

9. Teaching techniques *(Student centric techniques will be considered. The means for students to participate in defining their own study path, the identification of eventual fallbacks and the remedial measures that will be adopted in those cases will be described.)*

Course. Teaching is done interactively using a video projection system, powerpoint, virtual whiteboard and interactive programming in DevC++ together with students. The basic concepts are presented and then example problems are discussed. These are solved interactively, going through the stages of understanding the requirements, formalizing the algorithm, developing the code, correcting errors and validating the results. All course materials are available on the Moodle platform, in electronic format.

Laboratory. The laboratory is based on individual programming in the C/C++ language on the Moodle platform. Each student has an individual computer. Students have both solved and proposed problems at their disposal. Each exercise is included in an individual Virtual Programming Lab (VPL). The laboratory is preceded by short presentations to familiarize themselves with the theoretical concepts. All laboratory materials are available on the Moodle platform, in electronic format.

10. Contents

COURSE		
Chapter	**Content**	**No. hours**
1	Basics of C programming: features; creating, compiling and running a C program; structure of a C program; variables, constants; data types (fundamental types, type declarations, type and access modifiers, variable scope); assignment statements; operators and expressions; formatted input/output operations (read and write functions); examples, practical applications.	6
2	Conditional and repetitive instructions: conditional instructions and operators (if-else, switch-case, ?:); repetitive instructions (while, do-while, for); break and continue instructions; examples, practical applications.	6

3	Complex data types: data arrays; strings; structures and unions (general concepts); examples, practical applications.	6
4	Functions: generalities; defining and calling functions, prototypes; recursive functions; examples, practical applications.	8
5	Pointers and data files: defining and working with pointer variables; working with data files; examples, practical applications.	2
	Total:	28
Bibliography:		

LABORATORY		
Crt. no.	**Content**	**No. hours**
1	Description of the working environment; editing, compiling, executing programs.	2
2	Simple programs, variables and constants, formatted input/output operations and expressions.	2
3	Fundamental data types, operators and data type conversions.	2
4	Conditional instructions: decision and selection.	2
5	Loops with initial test, final test and counter. Flow control instructions.	2
6	Test #1	2
7	Arrays and strings. Multidimensional arrays.	6
8	Simple functions and recursion. Variable scopes. Modularization of code in multiple files.	6
9	Review problems.	2
10	Final lab test.	2
	Total:	28

Bibliography:
M. Dogariu, L.D. Ştefan, B. Ionescu, Platformă laborator, UPB, ETTI-EAII, Moodle https://curs.upb.ro/2021/course/view.php?id=8864;
C++, http://www.cplusplus.com (accesat 2022).

11. Evaluation

Activity type	11.1 Evaluation criteria	11.2 Evaluation methods	11.3 Percentage of final grade
11.4 Course	- knowledge of the fundamental theoretical notions of computer programming and in particular the C language; - solving programming problems using the C language;	Practical computer exam and interview. The topics cover the entire subject, creating a synthesis between theoretical coverage and solving programming problems.	50

11.5 Seminary/laboratory/project	- solving programming problems using the C language;	Practical evaluation, on the computer, along the way. Verification work;	20
	- solving programming problems using the C language;	Practical evaluation, on the computer, at the end of the laboratory. Final colloquium.	30
11.6 Passing conditions			
- participation in laboratory work; - accumulation of at least 50% of the score for the discipline (laboratory and exam).			

12. Corroborate the content of the course with the expectations of representatives of employers and representative professional associations in the field of the program, as well as with the current state of knowledge in the scientific field approached and practices in higher education institutions in the European Higher Education Area (EHEA)

The curriculum provides graduates with the knowledge necessary to understand the operating principle of a computing system as well as the basics of their programming using a high-level language such as C. The current technological progress of electronic and telecommunications devices is conditioned by the ability to develop and experiment using programming languages. Thus, the discipline of computer programming is fundamental in the training of future generations of engineers and researchers in the field;

The curriculum thus provides graduates with skills appropriate to the needs of current qualifications and a modern, quality and competitive scientific and technical training, which will allow them to be quickly employed after graduation. This is perfectly aligned with the policy of the Politehnica University of Bucharest, both in terms of content and structure, as well as in terms of the skills and international openness offered to students. Potential employers target both the academic environment (teaching and research profile) and the industrial research and development environment, such as organizations/companies of any size, from small ones created by students/master's students (e.g. start-ups and spin-offs), to multinational ones.

Date	Course lecturer	Instructor(s) for practical activities
25.09.2025	Prof. Dr. Bogdan Emanuel Ionescu, S.l./Lect. Dr. Mihai Dogariu	Prof. Dr. Bogdan Emanuel Ionescu

Date of department approval	Head of department

Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti

Facultatea de Electronică, Telecomunicaţii şi

Tehnologia Informaţiei

Date of approval in the Faculty Council	Dean
26.09.2025	Prof. Dr. Mihnea Udrea

