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	**Telecommunications**
1.4 Domain of studies	Electronic Engineering, Telecommunications and Information Technology
1.5 Cycle of studies	Bachelor/Undergraduate
1.6 Programme of studies	Technologies and Telecommunications Systems

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				Conf. Dr. Iulian Nastac			
2.3 Instructor for practical activities				Conf. Dr. Iulian Nastac			
2.4 Year of studies	1	2.5 Semester	1	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	3.3 seminary/laboratory	2
3.4 Total hours in the curricula	56	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	No prerequisites
4.2 Results of learning	No prerequisites

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

5.1 Course	The course will be held in a room with computer and projection system.
5.2 Seminary/ Laboratory/Project	The laboratory will be equipped with computers and adequate software. The attendance is mandatory (according to the current regulations imposed by UPB). Individual access to computers and integrated development environments for C/C++, as well as internet access (access to e-learning Moodle).

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)*

The discipline studies in its first part the general principles of structured programming, starting from the knowledge and understanding of the functioning of a computing system, and in the second part the fundamental problems of programming in C. The C language is presented in a gradual way, having covered all the essential aspects of C programming (data organization, instructions, vectors, functions, pointers).

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	Applying the fundamental knowledge, concepts and methods with respect to computer architecture, microcontrollers and programming languages.
Transversal (General) Competences	The capacity to get informed, with respect to scientific literature, which supports personal development as well as professional one.

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.* assimilate fundamental concepts of computer programming, assimilate basic programming concepts in C/C++, assimilate the knowledge neccesary to solve basic programming problems, assimilate the knowhow to debug a program in C/C++.
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).* ability to understand and explain a program written in C/C++ programming language, ability to design a simple algorithm, ability to validate the results of a program in C/C++, ability to identify programming solution, ability to communicate and justify solutions.

Responsability and autonomy	*The student's capacity to autonomously and responsably apply their knowledge and skills.* the capacity to select and analyze bibliographical sources, the capacity to propose and contribute with new solutions to problems, the capacity to learn new concepts, the capacity to communicate information to other peers, 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 both a video projection system for PowerPoint presentations, as well as a black board. The basic concepts are presented and then example cases are discussed. They are solved interactively, going through the steps of understanding requirements, formalizing the algorithm, developing, code, correcting errors and validating results. All course materials are available on the Moodle platform in electronics format. Laboratory: the laboratory relies on individual C/C++ preprogramming on the Moodle platform. Each student has an individual computer. Students have both solved and proposed problems. The laboratory is preceded by short presentations to familiarize with theoretical concepts. All laboratory materials are available on Moodle platform in electronic format.

10. Contents

COURSE		
Chapter	**Content**	**No. hours**
1	Introduction. General concepts. Computational systems: general information, way of functioning, dependencies between hardware and software. Types of programming languages.	4
2	Basics of C programming: characteristics; the creation, compilation and execution of a C program; the structure of a C program; variables, constants, data types (fundamental types of declarations, type and access modifiers, valability domain of variables); assignment; operators and expressions; input and output operations (reading and writing functions); examples and practical applications.	6
3	Conditional and repetitive instructions: instructions and conditional operators (if-else, switch-case, ?:); repetitive instructions (while, do-while, for); break and continue instructions; examples and practical applications.	6
4	Funtions: general concepts; defining and calling functions, prototypes; recursive functions; examples and practical applications.	4
5	Complex data types: matrices, strings of characters (general notions); examples and applications.	6
6	Pointers; defining and working with pointer variables; examples and applications.	2
	Total:	

Bibliography:

D.I. Năstac, Course Notes: Computers Programming and Programming Languages 1, UPB, ETTI, Moodle: https://curs.upb.ro/;

C++, http://www.cplusplus.com

D.I. Năstac, Programarea calculatoarelor în limbajul C – Elemente fundamentale, Editura Printech, Bucureşti, 2008.

LABORATORY		
Crt. no.	**Content**	**No. hours**
1	Presenting the work methodology; editing, compiling and program execution.	2
2	Basic programs, variables and constants, input and 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 and final condition, as well as counter; break and continue instructions.	4
6	Simple and recursive functions. Visibility domains for variables. Dividing the program into multiple files (modularization).	4
7	Matrices and character strings. Multidimentional vectors.	4
8	Pointers.	2
9	Recapitulative problems.	2
10	Project/Homework evaluation.	2
11	Final laboratory examn.	2
	Total:	28

Bibliography:

D.I. Năstac, Course Notes: Computers Programming and Programming Languages 1, UPB, ETTI, Moodle: https://curs.upb.ro/2023/course/view.php?id=10031;

C++, http://www.cplusplus.com

D.I. Năstac, Programarea calculatoarelor în limbajul C – Elemente fundamentale, Editura Printech, Bucureşti, 2008.

11. Evaluation

Activity type	11.1 Evaluation criteria	11.2 Evaluation methods	11.3 Percentage of final grade
11.4 Course	Knowledge of fundamental theoretical notions of computer programming and in particular C language; solving programming problems using C language	Written exam in the exam session corresponding to the semester. The topics cover the entire subject, a synthesis between theoretical background and programming problem solving.	50%
11.5 Seminary/laboratory/project	Problem solving using C language	Final laboratory exam	30%
	Proofing the functioning of written programs	Project evaluation with accent on the practical competences	20%
11.6 Passing conditions			
Mandatory laboratory presence Obtaining a minimum of 50% from the laboratory total number of point Obtaining a minimum of 50% from the score of 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 course curriculum provides graduates with the necessary knowledge to understand the principles of operation of a computing system and the basis for their programming using a high-level language such as the C language The current technological progress of electronic and telecommunications devices is conditioned by the ability to develop and experiment using programming languages. So the discipline of computer programming is fundamental in training 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 that allows them to engage rapidly after graduation. It is perfectly framed in the politics of the Politehnica University of Bucharest, both in terms of content and structure, as well as in terms of the international skills and openness offered to students. Potential employers target both academia (didactic and research profile) and industrial research and development environment such as organizations/firms of any size, from small ones created by students/master students (e.g. start-up and spin-off) to multinational ones.

Date	Course lecturer	Instructor(s) for practical activities
21.09.2026	Conf. Dr. Iulian Nastac	Conf. Dr. Iulian Nastac

Date of department approval	Head of department
22.09.2026	Conf. Dr. Serban Georgica Obreja

Date of approval in the Faculty Council	Dean
22.09.2026	Prof. Dr. Mihnea Udrea