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)				Structuri de date şi algoritmi Data structures and algorithms			
2.2 Course Lecturer				S.l./Lect. Dr. Mihai-Gabriel Constantin			
2.3 Instructor for practical activities				S.l./Lect. Dr. Mihai-Gabriel Constantin			
2.4 Year of studies	2	2.5 Semester	I	2.6. Evaluation type	E	2.7 Course regime	Ob
2.8 Course type		D	2.9 Course code	04.D.03.O.005		2.10 Tipul de notare	Nota

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

3.1 Number of hours per week	3.5	Out of which: 3.2 course	2.00	3.3 seminary/laboratory	1.5
3.4 Total hours in the curricula	49.00	Out of which: 3.5 course	28	3.6 seminary/laboratory	21
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.					45
Tutoring					0
Examinations					6
Other activities (if any):					0

3.7 Total hours of individual study	51.00
3.8 Total hours per semester	100
3.9 Number of ECTS credit points	4

4. Prerequisites (if applicable) (where applicable)

4.1 Curriculum	Computer programming and programming languages 1
4.2 Results of learning	General knowledge of the C language and computer programming.

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

5.1 Course	Video projection system for physical presence and access to the Microsoft Teams platform for online teaching. Attendance at the course is mandatory, according to ETTI regulations.
5.2 Seminary/ Laboratory/Project	Laboratory equipped with computing systems and access to the Moodle e-learning platform for physical presence and access to the Moodle e-learning platform and the Microsoft Teams platform for online teaching. Laboratory attendance is mandatory, according to ETTI regulations.

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: mastering the mechanisms of storing, structuring and processing data with complex composition. Study of the basic principles used in the composition of algorithms as an essential stage in the efficient development of software applications. Criteria for efficient program design. Case studies and methods for evaluating the performance of algorithms.

Laboratory: practical mastering, through the implementation of software programs, of the concepts taught in the course. Solving concrete practical problems that include elements of data structures and algorithms.

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.* • Ability to select and browse bibliographic sources, • Ability to promote and contribute with new solutions, • Ability to learn new concepts, • Ability to communicate information with other colleagues, • 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 VSCode 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	Pointers and data structures: working with pointer variables; dynamic memory allocation (using dynamic vs. static representations); working with structures; examples, practical applications.	6
2	Data lists: defining and working with data lists; examples, practical applications. Data queues: defining and working with queues; examples, practical applications. Data stacks: defining and working with stacks; examples, practical applications.	8
3	Data trees: generalities - addressing some types of applications through trees; defining and working with binary trees; examples, practical applications and sorting algorithms.	8

4	Interactive solving of various problems that involve algorithmization: starting from a statement made in a formal language, translating it into an algorithm and then into a functional program.	6
	Total:	28

Bibliography:
M.G. Constantin, Note de curs, Structuri de Date si Algoritmi, postate online, platforma Moodle
M.G. Constantin, Probleme asociate notelor de curs, Structuri de Date si Algoritmi, postate online, platforma Moodle
M. Dogariu, M.G. Constantin, L.D. Stefan, Platforma de laborator Structuri de Date şi Algoritmi, Structuri de Date si Algoritmi, postate online, platforma Moodle
„CPP Coding Standards - 101 Rules Guidelines and Best Practices”, Herb Stutter, Andrei Alexandrescu.

LABORATORY

Crt. no.	Content	No. hours
1	Pointers	2
2	Data structures	2
3	Algorithmization of problems with data structures	2
4	Singly linked lists	2
5	Algorithmization of problems with singly linked lists	2
6	Circular, doubly linked lists, stacks, queues	2
7	Algorithmization of problems with complex lists	2
8	Binary trees	2
9	Algorithmization of problems with binary trees	2
10	Recap	2
11	Lab colloquium	1
	Total:	21

Bibliography:
M.G. Constantin, Note de curs, Structuri de Date si Algoritmi, postate online, platforma Moodle
M.G. Constantin, Probleme asociate notelor de curs, Structuri de Date si Algoritmi, postate online, platforma Moodle
M. Dogariu, M.G. Constantin, L.D. Stefan, Platforma de laborator Structuri de Date şi Algoritmi, Structuri de Date si Algoritmi, postate online, platforma Moodle
„CPP Coding Standards - 101 Rules Guidelines and Best Practices”, Herb Stutter, Andrei Alexandrescu.

11. Evaluation

Activity type	11.1 Evaluation criteria	11.2 Evaluation methods	11.3 Percentage of final grade
11.4 Course	The ability to abstract a real problem and implement an effective solution in the C/C++ language	Final exam, held virtually on the Moodle platform. The student solves the proposed problems and their solution and proposed solutions are analyzed.	50

11.5 Seminary/laboratory/project	The ability to solve a programming problem in the C/C++ language using advanced data structures	Intermediate laboratory work, practically supported on the Moodle platform. The student solves the proposed requirements and the way to solve them and the proposed solutions are analyzed (week 7)	20
	Verification of knowledge acquisition from the course/laboratory	Laboratory colloquium, held virtually on the Moodle platform. The student solves the proposed problems and their solution and proposed solutions are analyzed (week 14)	30
11.6 Passing conditions			
- participation in laboratory work; - accumulation of at least 50% of the score for the laboratory; - 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 course curriculum provides graduates with the ability to algorithmize a computational problem of medium complexity, knowledge of classical algorithms for solving computational problems, the ability to transform a computational problem expressed in natural language into a programming language and solution. The technological progress of electronic and telecommunications devices is including the ability to develop and experiment with current programming languages. Thus, the discipline of computer programming is fundamental in the training of future generations of engineers and researchers in the field.

The program thus provides graduates with skills appropriate to current qualification needs and a modern, quality and competitive scientific and technical training, which will allow them to be employed quickly after graduation. It 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 students (e.g. start-ups and spin-offs), to multinational ones.

Date	Course lecturer	Instructor(s) for practical activities
21.09.2026	S.l./Lect. Dr. Mihai-Gabriel Constantin	S.l./Lect. Dr. Mihai Dogariu

Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti

Facultatea de Electronică, Telecomunicaţii şi

Tehnologia Informaţiei

S.l./Lect. Dr. Mihai-Gabriel Constantin



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

