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 2 Computer programming and programming languages 2			
2.2 Course Lecturer				S.l./Lect. Dr. Mihai Dogariu			
2.3 Instructor for practical activities				S.l./Lect. Dr. Mihai Dogariu			
2.4 Year of studies	1	2.5 Semester	II	2.6. Evaluation type	E	2.7 Course regime	Ob
2.8 Course type		F	2.9 Course code	04.F.02.O.011		2.10 Tipul de notare	Nota

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

3.1 Number of hours per week	3	Out of which: 3.2 course	2.00	3.3 seminary/laboratory	1
3.4 Total hours in the curricula	42.00	Out of which: 3.5 course	28	3.6 seminary/laboratory	14
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.					52
Tutoring					0
Examinations					6
Other activities (if any):					0

3.7 Total hours of individual study	58.00
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	Solving low-difficulty problems in the C/C++ language.

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

5.1 Course	Video projection system for course presentation and access to the Moodle platform for exemplifying the solution of some exercises.
5.2 Seminary/ Laboratory/Project	Laboratory equipped with computing systems and access to the Moodle e-learning platform. Attendance at the laboratory 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 abstraction of programming problems in the form of classes and objects. Understanding the interaction between objects and the relationships between different classes. Using the notions of object-oriented programming to solve problems expressed in natural language.

Laboratory: practical mastering, through the implementation of software programs, of the notions taught in the course. Solving concrete practical problems that include elements of object-oriented programming.

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	- Understanding and using fundamental concepts in the field of communications and information transmission. - Applying fundamental knowledge, concepts and methods regarding the architecture of computer systems, microcontrollers, programming languages and techniques. - Designing and using computer systems and computer networks.
Transversal (General) Competences	- The ability to make decisions in order to solve current or unpredictable problems that arise in the process of operating computer systems. - The ability to constantly inform and document oneself for personal and professional development by reading specialized literature. - The ability to communicate and present technical content in both Romanian and English.

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.* • The ability to make decisions in order to solve current or unpredictable problems that arise in the process of operating computer systems. • The ability to constantly inform and document oneself for personal and professional development by reading specialized literature. • The ability to communicate and present technical content in both Romanian and English. • Lists the principles of object-oriented programming. • Defines domain-specific notions, such as: class, object, inheritance, polymorphism. • Describes the different access specifiers and their implications. • Correctly uses the studied operators and their overloaded variants. • Interprets and implements classes based on textual descriptions formulated in natural language. • Describes the relationships between objects in the context of inheritance and polymorphism. • Interprets and defines classes based on UML diagrams.
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).* • Selects and groups information regarding the structure of a class formulated in natural language. • Uses the principles of object-oriented programming in an argumentative manner. • Uses appropriate scientific language. • Experimentally verifies identified solutions. • Solves practical applications. • Interprets causal relationships appropriately. • Analyzes and compares the results obtained with the desired ones. • Argues for the identified solutions/solution methods.
Responsability and autonomy	*The student's capacity to autonomously and responsably apply their knowledge and skills.* • Selects appropriate bibliographic sources and analyzes them. • Respects the principles of academic ethics, correctly citing the bibliographic sources used. • Demonstrates receptivity to new learning contexts. • Demonstrates autonomy in organizing the learning situation/context or the problem situation to be solved.

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.)*

Starting from the analysis of the students' learning characteristics and their specific needs, the teaching process will explore both expository (lecture, exposition) and conversational-interactive teaching methods, based on discovery learning models facilitated by direct and indirect exploration of reality (experiment, demonstration, modeling), but also on action-based methods, such as exercise, practical activities and problem solving.

Lectures will be used in the teaching activity, based on Power Point presentations that will be made available to students. Each course will begin with a recap of the chapters already covered, with an emphasis on the concepts covered in the last course.

The presentations use images, diagrams and source code examples, so that the information presented is easy to understand and assimilate.

10. Contents

COURSE

Chapter	Content	No. hours
1	Introduction: general aspects of object-oriented programming, migration from procedural programming language (C) to object-oriented programming language (C++), namespaces	1
2	Classes & objects: introduction to class and object concepts, access specifiers, constructors, initialization lists, destructors, object vectors, special member functions, separation of classes into files, keywords 'static', 'friend', solving exercises.	9
3	Inheritance: basic concepts, order of calling constructors/destructors, hiding members, overriding functions, virtuality, keywords 'final', 'override', different types of inheritance, solving exercises.	6
4	Operator overloading: assignment, extraction/insertion from/into streams, increment/decrement, binary arithmetic, relational, solving exercises	2
5	Templates: generic programming principles, class templates, function templates, specializations, polymorphism.	2
6	Standard Template Library: containers (sequence, associative, unordered associative, container adapters), iterators, algorithms, exercise solving.	4
7	Working with files: input/output streams, examples.	2
8	Final verification.	2
	Total:	28

Bibliography:

1. Mihai DOGARIU, Programare Obiect-Orientată, suport de curs electronic, https://curs.upb.ro/2021/course/view.php?id=8975
2. “The C++ Programming Language”, 4th Edition, Bjarne Stroustrup
3. „CPP Coding Standards - 101 Rules Guidelines and Best Practices”, Herb Stutter, Andrei Alexandrescu.
4. „Head First Object-Oriented Design and Analysis”, Brett McLaughlin, Gary Pollice, David West.
5. „Head First Design Patterns”, Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra.

LABORATORY

Crt. no.	Content	No. hours
1	Classes & objects: introduction to class and object concepts, access specifiers, constructors.	2
2	Classes & objects: initialization lists, destructors, object vectors.	2
3	Lab test #1	2
4	Inheritance: basic concepts, UML diagram, order of calling constructors/destructors, hiding members.	2
5	Virtuality: function overriding, virtual destructors, pure virtual functions	2
6	Operator overloading, Standard Template Library	2
7	Lab test #2	2
	Total:	14

Bibliography:
Mihai DOGARIU, Programare Obiect-Orientată, suport de laborator electronic, https://curs.upb.ro/2021/course/view.php?id=8975

11. Evaluation

Activity type	11.1 Evaluation criteria	11.2 Evaluation methods	11.3 Percentage of final grade
11.4 Course	The ability to solve a programming problem in the C/C++ language using object-oriented programming concepts.	Final exam, held virtually on the Moodle platform. The student solves the proposed requirements and their solution and proposed solutions are analyzed (session).	50
11.5 Seminary/laboratory/project	The ability to solve a programming problem in the C/C++ language using object-oriented programming concepts.	Intermediate laboratory work, practically supported on the Moodle platform. The student solves the proposed requirements and their solution and proposed solutions are analyzed (weeks 5&6).	20
	The ability to solve a programming problem in the C/C++ language using object-oriented programming concepts.	Final laboratory work, practically supported on the Moodle platform. The student solves the proposed requirements and their solution and proposed solutions are analyzed (weeks 13&14).	30
11.6 Passing conditions			
- Participation in laboratory work. - Accumulation of at least 50% of the laboratory score. - Accumulation of at least 50% of the subject score.			

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 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
25.09.2025	S.l./Lect. Dr. Mihai Dogariu	S.l./Lect. Dr. Mihai Dogariu

Date of department approval	Head of department

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