Universitatea Națională de Știință și Tehnologie Politehnica București

Facultatea de Electronică, Telecomunicații și

Tehnologia Informației

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 și limbaje de programare 3 - Proiect Python Computer programming and programming languages 3 - Python Project
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	2	2.5 Semester	I	2.6. Evaluation type	V	2.7 Course regime	Op
2.8 Course type	F	2.9 Course code	04.F.03.A.009			2.10 Tipul de notare	Nota

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

3.1 Number of hours per week	1	Out of which: 3.2 course	0.00	3.3 seminary/laboratory	1
3.4 Total hours in the curricula	14.00	Out of which: 3.5 course	0	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.	9
Tutoring	0
Examinations	2
Other activities (if any):	0

3.7 Total hours of individual study	11.00
3.8 Total hours per semester	25
3.9 Number of ECTS credit points	1

4. Prerequisites (if applicable) (where applicable)

4.1 Curriculum	• Computer Programming and Programming Languages 1 • Computer Programming and Programming Languages 2
4.2 Results of learning	• General knowledge of computer programming.

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

5.1 Course	Not applicable.
5.2 Seminary/ Laboratory/Project	Video projection system for physical presence and access to the Microsoft Teams platform for online consultations. Attendance at project meetings 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)*

- Training skills in using the Python language in application development.
- Understanding fundamental programming concepts and applying them to real projects.
- Developing the ability to analyze, design and implement software solutions.
- Practicing teamwork and presenting software projects.

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 basic knowledge, concepts and methods regarding computer system architecture, microprocessors, microcontrollers, programming languages and techniques
Transversal (General) Competences	Methodical analysis of problems encountered in the activity, identifying elements for which there are established solutions, thus ensuring the fulfillment of professional tasks. Adaptation to new technologies, professional and personal development, through continuous training using printed documentation sources, specialized software and electronic resources in Romanian and, at least, in an internationally spoken language.

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.* • learning to program in the Python programming language • learning to program a graphical interface in Python • learning to integrate databases and external APIs into a Python application • learning to debug a program in the Python 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 Python code, • the ability to use external APIs in Python, • the ability to identify programming solutions, • the ability to understand the operational flow and debug programs in Python, • 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, • Developing 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.)*

The project is based on team programming of an application in the Python language. Each team chooses a theme, assigns each member a specific task, develops a graphical interface and integrates all the developed submodules into a final application, using concepts of object-oriented programming, databases and the use of external APIs.

10. Contents

PROJECT		
Crt. no.	**Content**	**No. hours**
1	Introduction to Python. Data types, operators, control structures, simple functions. Examples: mini-calculator, unit conversions.	2
2	Defining project themes. Choosing the theme and establishing requirements. Examples: chatbot, budget app, recipe organizer, memory game.	2
3	Application structure and backend. Implementation of basic functionalities, initial testing in command line.	2

4	Interface and OOP. Creating the graphical interface, connecting with the backend, introduction to OOP (classes, inheritance).	2
5	Advanced functionalities and databases. Data persistence, database connectivity, integration of external APIs.	2
6	Optimization and testing. Debugging, code optimization, preparation of the final project.	2
7	Final evaluation. Presentation of projects and discussions on implemented solutions.	2
	Total:	14
Bibliography:		

11. Evaluation

Activity type	11.1 Evaluation criteria	11.2 Evaluation methods	11.3 Percentage of final grade
11.4 Course			
11.5 Seminary/laboratory/project	The ability to successfully model and solve a problem programmatically in Python.	The evaluation is based on activity during the semester and laboratory attendance.	50
	Final verification of the project and documentation.	Final evaluation of the Python project.	50
11.6 Passing conditions			
Obtaining 50% of the total 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)

In recent years, Python has become the most popular programming language, representing the pillar of the development of artificial intelligence. In addition, there are numerous other applications related to statistics, big data, database processing or social media application development in which Python is the most commonly used programming language. This project complements the preparation of electronics students for the workplace. Based on the knowledge gained from this discipline, students can perform a multitude of tasks in this programming language, from specifying requirements to executing, debugging, and interpreting results.

Date	Course lecturer	Instructor(s) for practical activities
25.09.2025		Prof. Dr. Bogdan Emanuel Ionescu

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