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	**Electronic Devices, Circuits and Architectures**
1.4 Domain of studies	Electronic Engineering, Telecommunications and Information Technology
1.5 Cycle of studies	Bachelor/Undergraduate
1.6 Programme of studies	Microelectronics, Optoelectronics and Nanotechnologies

2. Date despre disciplină

2.1 Course name (ro) (en)				Programarea calculatoarelor şi limbaje de programare 3 - Proiect Python Computer Programming and Programming Languages 3 - Python Project			
2.2 Course Lecturer				Prof. Dr. Razvan Craciunescu			
2.3 Instructor for practical activities				Prof. Dr. Razvan Craciunescu			
2.4 Year of studies	2	2.5 Semester	1	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	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. Object oriented programming.

4.2 Results of learning	Basic requirements of installing programs, using editors to write source code and making computer programs.

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

5.1 Course	The session will be held in a room equipped, preferably, with a video projector and internet access (to allow simultaneous use in the form of a Teams video conference). Students may work on their own computers.
5.2 Seminary/ Laboratory/Project	Attendance of sessions is mandatory (as stated by the regulations for undergraduate university studies). Students may work on their own computers during the sessions in rooms with internet access.

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 has the general objective of familiarizing students with the concept of the Python programming language, as well as connex modern technologies often used in practice.

During the project hours, the students will become familiar with the Python programming language, on the student's computer. The study of the Python language will combine teaching notions with solving exercises by the students, individually, on their own computers, identifying and exercising abilities necessary to computer model situations from reality.

Based on the knowledge gained from this discipline, the future electrical engineer will be able to implement or modify programs or command-line scripts specific to modern, scriptable and automatable software development and testing activities, being able to create programs from requirements specification to execution, debugging and interpretation of results.

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	**C3.** Application of basic knowledge, concepts and methods regarding the architecture of computer systems, microprocessors, microcontrollers, programming languages and techniques.
Transversal (General) Competences	**CT1.** Methodical analysis of the problems encountered in the activity, identifying the elements for which there are established solutions, thus ensuring the fulfillment of professional tasks. **CT3.** Adapting 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 one language of international circulation.

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.* Is able to answer a series of questions based on a portfolio of activities carried out during the semester, as part of an oral examination. Describes the Python program needed to solve a given problem. Highlights the need to use certain modules. Describes Python language concepts needed to solve a given problem.
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).* Designing, programming, debugging and successfully running a program to solve a given problem. Implementing a program with specified facilities, in the Python language. Checking and debugging the validity of a given program. Working with corresponding basic data types (such as mutable/immutable concept), case sensitiveness, data display and structures (lists, strings, dictionaries, tuples, etc.). Highlighting the utility of the slice operator. Using flow control statements, Boolean expressions, functions, modules, appropriately to solve problems. Identifying situations that benefit from the use of object-oriented programming concepts in Python.
Responsability and autonomy	*The student's capacity to autonomously and responsably apply their knowledge and skills.* Demonstrating responsiveness to new learning contexts. Conspecting in advance the course materials, laboratory, to the extent that they are made available. In case of absence, going through the material taught, made available, by oneself. Solving homework individually, autonomously, respecting academic ethics. Respecting the principles of academic ethics, individually carrying out the activities marked in this sense, also correctly citing the bibliographic sources used, if the situation requires it. Applying principles of professional ethics/deontology in the analysis of the technological impact of the solutions proposed in the specialized field on the environment.

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 didactic materials used are the session notes and presentations, also available in electronic format.

Starting from the analysis of the students' learning characteristics and their specific needs, the teaching process will explore both expository (lecture, exposition), problem-solving and conversational-interactiveteaching methods, based on action-based learning models, such as exercise, practical activities and problem solving. Interactivity with students through the associated applied activities. Intervals are reserved for presentation, analysis and solving of some practical problems (reality modeling).

In the applied section, teaching is based on the use of the expository method (covering the communication and demonstrative function). The dialogue during the course is also extended during the application sessions. These are necessary to prepare students for homework and verification tests along the way.

Feedback will also be used, as a way of adapting the pedagogical approach to the students' learning needs.

10. Contents

PROJECT

Crt. no.	Content	No. hours
1	Introductory information. Installing Python on Windows and/or Linux. Using a Python interpreter. Running Python code files.	2
2	Fundamental data types. Mutable/immutable. Naming variables. Case sensitive. Displaying data. Exercises.	2
3	Data structures. Lists. Tuples. Dictionaries. The slice operator. Exercises.	2
4	Execution flow control instructions. Boolean expressions. Functions. Modules. Exercises.	2
5	Concepts of Object Oriented Programming in Python. Classes. Attributes. Methods. Inheritance. Exercises.	2
6	Applications. Visual interfaces (GUI). Native interfaces via tkinter. Venv. Web interfaces via Flask. Exercises.	2
7	Final verification	2
	Total:	14

Bibliography:

V.G.Voiculescu, Python project, electronic suport , https://ctipub.sharepoint.com/:f:/s/ProiectPython-Voiculescu/EoWmoTyZ3idIj26kq8M30pgBFtbztZBxs0O5n5Nl9O-WXA?e=LLStlO
W3schools, Python 3, https://www.w3schools.com/python/, 2024
D. Beazley, B.K. Jones, Python cookbook: Recipes for mastering Python, 3rd edition, O'Reilly, 2013

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	Correct identification of the theoretical and practical application in contexts of the studied concepts from Python as well as the features studied for them. Ability to model and successfully solve a problem programmatically in Python.	The practical activity is constantly checked throughout the semester. Homework. Final evaluation based on the project.	100%
11.6 Passing conditions			
Implementation of a program with specified facilities in Python language, modeling, programming, running successfully in order to get the solution. Obtaining 50% of the total score or the minimum score provided by the regulation.			

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)

Python is a very popular language, being the basis of some intensively used social media applications, also having applicability in future fields such as artificial intelligence, data science. We are in a process of adoption this programming language at the faculty level, but each discipline describes it only minimally, in order to introduce the specific study concepts. It is a popular language and in some scripting, testing, building frameworks found in the target fields of our graduates, and with the help of this subject the future electrical engineer will be able to implement or modify programs or command line scripts specific to development and testing activities modern, scriptable and automatable software.

Based on the knowledge gained from this course, the future electronics engineer will be able to implement or modify programs or command-line scripts specific to modern, scriptable and automatable software development and testing activities, being able to create programs from requirements specification to execution, debugging and interpretation of results.

Date	Course lecturer	Instructor(s) for practical activities
21.09.2026		Prof. Dr. Razvan Craciunescu

Date of department approval	Head of department
22.09.2026	Prof. Dr. Claudius Dan

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