

SYLABUS

Approved,
DEAN,
Prof.dr. ing. Gilbert-Rainer GILLICH

1. Program data

1.1. Higher education institution	Universitatea Babes Bolyai Cluj-Napoca
1.2. Faculty	Faculty of Engineering
1.3. Department	Department of Engineering Sciences
1.4. Field of study	Electrical Engineering
1.5. Cycle of studies	Bachelor
1.6. Study program	Applied Informatics in Electrical Engineering

2. Discipline data

2.1. Name of the discipline	COMPUTER PROGRAMMING AND PROGRAMMING LANGUAGES II						
2.2. The holder of the course activities	Ş.I.dr.ing. Cornelia-Victoria Anghel-Drugărin						
2.3.1. Holder of the seminar activities	-						
2.3.2. Holder of laboratory activities	Ş.I.dr.ing. Cornelia-Victoria Anghel-Drugărin						
2.3.3. Holder of the project activities	-						
2.4. Year of study	1	2.5. Semester	2	2.6. Type of assessment	E	2.7. Discipline regimen	Obl.

3. Estimated total time (hours per semester of teaching activities)

3.1. Number of hours per week	5	of which: 3.2. course	2	3.3. laboratory	3
3.4. Total hours of the curriculum	70	of which: 3.5. course	28	3.6. laboratory	42
Distribution of the time fund					55Hours
Study by textbook, course support, bibliography and notes					15
Additional documentation in the library, on specialized electronic platforms and in the field					15
Preparation of laboratories/ projects, themes, program lines, applications					23
Tutoring					-
Examination					2
Other activities:					YES
3.7. Total hours of self-study	55				
3.8. Total hours per semester	125				
3.9. Number of credits	5				

4. Preconditions (where applicable)

4.1. of curriculum	<ul style="list-style-type: none"> Applied informatics, PCLP_I; TIC
4.2. of competencies	<ul style="list-style-type: none"> Digital competencies

5. Conditions (where applicable)

5.1. of the course	<ul style="list-style-type: none"> PC/laptop, Internet network access, UBB account, Office 365 account, Microsoft TEAMS platform, Python programming language
5.2.1. of the seminar	<ul style="list-style-type: none"> -
5.2.2. of the laboratory	<ul style="list-style-type: none"> Personal computer, Internet network access, online UBB account, on the Microsoft TEAMS platform; Install free kit used programming language: Python
5.2.3. of the project development	<ul style="list-style-type: none"> -

6. Specific competences acquired

Professional skills	<ul style="list-style-type: none"> Proper application of knowledge of the fundamental disciplines mathematics, computer science, physics in the field of electrical engineering (CP 1). Proper use of fundamental concepts in computer science and information technology (CP 2). Proper use of fundamental concepts in the field of electrical engineering (CP 3).
Transversal competences	<ul style="list-style-type: none"> Establishing the objectives to be achieved, the conditions of implementation, the stages of work and the deadlines for completion. (CT 1). Efficient use of information sources, communication resources and assisted training (internet, specialized software applications, dedicated databases, on-line courses, etc.) in Romanian language and/or in an international language. (CT 4).

7. The objectives of the discipline (based on the grid of specific competences accumulated)

7.1. The general objective of the discipline	Acquiring the competence of computer programmer of future engineers.
7.2. Specific objectives: Objectives of knowledge (O.k.)	Elaboration of problem solving algorithms, the necessary steps to be taken to find the solution, by programming in Python of technical and engineering problems. Familiarization with the use of the facilities of a programming environment and of some programs for calculating and solving specific problems in the field of engineering, implementing them in Python codes and troubleshooting the executed programs.
Objectives of attitude (O.at)	Developing a positive attitude towards individual work and teamwork. Developing the attitude of assuming and empowering the role of self-evaluation and collegial evaluation.
Objectives of enabling (O.e)	Realization of practical PYTHON programming applications, in chourslation with the requirements of employers on the labor market.

8. Content

8.1. Course	Teaching methods	Observations
1. General notions of Python programming. Python scopes. Installation, interpreter , useful links. The first Python program. Lexical elements. Types of data. Operators. Priority of operators. Examples.	Presentation of the content of the discipline sheet and bibliography . Lecture, lecture, explanations, interactive discussions and dialogue with students. Examples of programs	2 hours
2. Variables in the Python programming language. Definitions. Statements. Local and global variables. Data operations in Python. Numbers. Complex numbers. Exchanging values. Python programs	Lecture, presentation/explanations, discussions and dialogue with students.	2 hours
3. Linear programming. Linear structure. Assignment operator. Problems.		2 hours
4. Functions in Python. Forwarding arguments. Lambda function. Apps		2 hours
5. Structured programming. Strings. Functions for manipulating strings. Comparison of strings. Fundamental control structures. If decision instruction. IF statement and IF-ELSE respectively. Operators used. Boolean values. Logical operators. Nested decision structures. Python applications.	Compiling a summary journal. We have proposed/foreseen various ways to come into contact with information (inductive teaching methods: problem-based learning).	2 hours
6. Procedural programming Lists. Functions and methods of working with lists. Dynamic lists. List operations: sorting, recursion, searching, etc. Operators "+" and "*" in and not in Practical examples.	Edifying practical examples of program code lines	2 hours
7. Repetitive control instructions in the Python programming language. Repetitive instruction FOR. Iterable data types. examples of LOOPS FOR. The range function to create iterable objects in the for loop. Batteries. Augmented assignment operators. Input validation loops. Nested curls. Examples of Python programs	Brainstorming Method We have provided situations in which the student can reflect on how he learned - heuristic discussions, fixing the acquired knowledge.	2 hours
8. The WHILE loop instruction. Programs. Interrupt instructions: break, continue, clauses to exit the loops. Examples of applicative programs with loops and interruptions in Python.		2 hours
9. Modular programming What is a Python module, variable range, packages, standard modules, module distribution. Generation of random modules - random mode. Modulation with functions. Mathematical module. The value of math.pi.	Exposure problematization, heuristic conversation, detailed explanation, challenge of students to	2 hours

Calendar module. Examples.	dialogue, discussions, fixation of acquired notions.	
10. Graphical interfaces in Python. The turtle module. Graphic objects	Exposition, problematization, heuristic conversation, detailed explanation, challenge of students to dialogue, discussions Program debugging We have proposed/foreseen tasks related to the future profession, relevant examples from professional practice	2 hours
11.Tupluri. Sets of crowds. Dictionaries. Operations with tuples. Useful functions and working methods. Operations with crowds. Frozenset. Dictionaries: working methods, delete/clear dictionary. Get() method. Change and add data pairs.		2 hours
12. Files and exceptions. File types. Methods of accessing files. Write/read/concatenate/add data to the file. Binary files. Exceptions.		2 hours
13. Object-oriented programming. Encapsulation of data. Objects and classes. The mechanism of inheritance. Handling exceptions. Examples of programs.		2 hours
14. Testing and inspection of programs • Black box testing, white box testing • Unit testing, integration testing • Inspection program: coding style, refactoring	Lecture, explanation, recap, summary journals. We have foreseen situations in which the student can reflect on how he has learned and acquired programming knowledge.	2hours
Bibliography 1. Frentiu, M., H.F. Pop, Serban G., Programming Fundamentals, Cluj University Press, 2006 2. The Python language reference. http://docs.python.org/py3k/reference/index.html 3. The Python standard library. http://docs.python.org/py3k/library/index.html 4. The Python tutorial. http://docs.python.org/tutorial/index.html 5. Vlad Tudor-Curs de programare Python3, Ed. InfoBits.ro		
8.2.2. Laboratory	Teaching methods	Observations
1. PSI training rules. Python installation. Programs applied to data types, operators and expressions	We have proposed/foreseen various ways to get in touch with information (projects, case studies, learning through discovery, etc.) Programs Applications, Practical examples We have proposed/foreseen various situations of teamwork. Development of the portfolio of programs and grids for evaluation.	3hours
2. Constants and variables in Python. Programs		3hours
3. Python linear programming		3hours
4. Functions in Python. Apps		3hours
5. Structured programming. programs with if and if-else decision instructions.		3hours
6. Procedural programming. Lists and operations with lists.		3hours
7.Python programs with FOR repetition instruction		3hours
8.Python programs with the while looping instruction. Interruption clauses		3hours
9.Modular programming		3hours
10. Graphical interfaces in Python. The turtle module.		3hours
11. Tupluri. Sets of crowds. Dictionaries.		3hours
12. Files and exceptions.		3hours
13. Object-oriented programming.		3hours
14.Testing and debugging python programs final discussions. End of laboratory work, teaching homework.		3hours
Bibliography 1.Doru Atanasiu Popescu- <i>Python - Notiuni fundamentale, culegere de probleme</i> editura infoBits.ro 2. https://www.pythonisti.ro/ 3. https://www.w3schools.com/python/		

9. Corroboration of the contents of the discipline with the expectations of the representatives of the epistemic community, professional associations and representative employers in the field related to the program

The content of the discipline was established, after consultation with the main employers in the field of engineering, the discussions with the committee for drawing up and reviewing the curriculum, in force, as well as with the

colleagues who teach related subjects.

Together with the students, we participate in external visits and / or teaching trips, at specialized companies in the county and region, in events such as: computer science circle, projects, round tables, workshops, Summer School, national and even international student scientific symposiums, which are related to the field of training.

We have proposed/foreseen tasks related to the future profession, relevant examples from professional practice.

10. Evaluation

Activity Type	10.1. Assessment criteria	10.2. Valuation methods	10.3. Weighting of the final grade
10.4. Course	Participation in debates	Number of interventions	10 %
	Level of knowledge gained	Grid test + Python application	60 %
10.5.1. Seminar	Involvement in activities		
	Level of competences acquired		
10.5.2. Laborator	Involvement in activities	Number of interventions Accomplished themes	10 %
	Level of practical competences acquired	Practical test, programming applications	20 %
10.5.3. Project	Promptness of going through the stages of the project		
	Quality of the project		
10.6. Minimum performance standard			
We have provided evaluations (formative) along the way (one in the course and two evaluation papers at the laboratory) in order to be able to discuss based on them a self-directed learning plan <ul style="list-style-type: none"> Promotion of laboratory applied activities with a minimum grade of 5 (five); Brief treatment of the subjects, the exam being considered to be passed, if the final grade calculated is a minimum of 5(five). 			

Date of completion

05.05.2022

Signature of the course holder

**Ș.I.dr.ing. Cornelia-Victoria
ANGHEL DRUGĂRIN**

Signature of the ~~seminar~~ holder / laboratory

**Ș.I.dr.ing. Cornelia-Victoria ANGHEL
DRUGĂRIN**

Date of approval in the department

.....

Signature of the DSI department director

Ș.I.dr.fiz. Hațiegan Cornel