

SYLLABUS

Algorithms and Programming

University year 2025-2026

1. Information regarding the programme

1.1. Higher education institution	Babes-Bolyai University
1.2. Faculty	Faculty of Psychology and Educational Sciences
1.3. Department	Department of Psychology
1.4. Field of study	Psychology - Cognitive Sciences
1.5. Study cycle	Bachelor level
1.6. Study programme/Qualification	Psychologist
1.7. Form of education	

2. Information regarding the discipline

2.1. Name of the discipline	Algorithms and Programming				Discipline code	MLE5119	
2.2. Course coordinator							
2.3. Seminar coordinator							
2.4. Year of study	1	2.5. Semester	1	2.6. Type of evaluation	C	2.7. Discipline regime	DS

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	4	of which: 3.2 course	2	3.3 seminar/laboratory	1+1
3.4. Total hours in the curriculum	56	of which: 3.5 course	28	3.6 seminar/laborator	28
Time allotment for individual study (ID) and self-study activities (SA)					hours
Learning using manual, course support, bibliography, course notes (SA)					12
Additional documentation (in libraries, on electronic platforms, field documentation)					12
Preparation for seminars/labs, homework, papers, portfolios and essays					10
Tutorship					6
Evaluations					4
Other activities:					
3.7. Total individual study hours					44
3.8. Total hours per semester					100
3.9. Number of ECTS credits					4

4. Prerequisites (if necessary)

4.1. curriculum	-
4.2. competencies	-

5. Conditions (if necessary)

5.1. for the course	Classroom with at least 80 seats, computer and video projector
5.2. for the seminar /lab activities	Room with at least 40 seats, computer and video projector / Laboratory needs at least 10 spare laptops/computers for students who do not have personal laptop

6.1. Specific competencies acquired ¹

¹ One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

Professional/essential competencies	<ul style="list-style-type: none"> Knowledge and understanding • Definition and description of programming paradigms and of language specific mechanisms, as well as identification of syntactic and semantic differences. • Description of existing software applications, on different levels of abstraction using adequate basic knowledge. Explanation and interpretation • Elaboration of adequate source code and testing of components in a well known programming language, based on given specifications, for basic programs and scripts.
Transversal competencies	<ul style="list-style-type: none"> • Application of efficient and rigorous working rules, manifest responsible attitudes towards the scientific and didactic fields, underlying the individual potential and respecting professional and ethical principles. • Use of efficient methods and techniques for learning, information, research and development of abilities for knowledge exploitation, for adapting to the needs of a dynamic society and for communication in a widely used foreign language.

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	<ul style="list-style-type: none"> • To know the basic concepts of software engineering (design, implementation and maintenance) and to learn Python programming language
7.2 Specific objective of the discipline	<ul style="list-style-type: none"> • To know the key concepts of programming • To know the basic concepts of software engineering • To gain understanding of basic software tools used in development of programs • To learn Python programming language and tools to develop, run, test and debug programs • To acquire and improve a programming style according to the best practical recommendations

8. Content

8.1 Course	Teaching methods	Remarks
1. Introduction to basic programming concepts and the Python programming language	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
2. Control flow concepts: if, while and for.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
3. Introduction to data structures: lists in Python.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
4. Methods for solving simple problems with lists (such as searching, aggregation or selection problems).	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
5. Procedural programming: introduction to functions in Python.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
6. Data structures II: dictionaries in Python. Methods for solving simple problems with dictionaries and lists.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
7. Nested data structures. Working with	Lecture, demonstrative	

datasets.	example, synthesis of knowledge, guided discovery	
8. Modular programming: Python modules and programming guidelines.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
9. Algorithm design: basic searching and sorting algorithms.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
10. File operations: reading and writing to text files in Python.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
11. Working with CSV datasets. Data preprocessing techniques.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
12. Design and implementation of applications of medium complexity, involving datasets and command line interface, based on given specifications.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
13. Recapitulation. Preparation for written examination.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
14. Written Examination	Written Exam	
Bibliography: Python like you mean it, by Ryan Soklaski – module 2. https://www.pythonlikeyoumeanit.com/module_2.html The Python Tutorial – parts 2.1.2, 3, 4, 5, 6 (just introduction, no sections), 7. https://docs.python.org/3/tutorial/		
8.2 Seminar / laboratory	Teaching methods	Remarks
1. Basic exercises with command line input output and simple expressions.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
2. Using control flow. Command line user interface applications.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
3. Creating and using functions.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
4. Problems using datasets and data processing.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
5. Reading datasets from file, saving results to file.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
6. Creating complex scripts and command line applications.	Lecture, demonstrative example, synthesis of knowledge, guided discovery	
7. Practical examination.	Practical examination.	
Bibliography Python like you mean it, by Ryan Soklaski – module 2. https://www.pythonlikeyoumeanit.com/module_2.html The Python Tutorial – parts 2.1.2, 3, 4, 5, 6 (just introduction, no sections), 7. https://docs.python.org/3/tutorial/		


9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- The course respects the IEEE and ACM Curricula Recommendations for Computer Science studies.
- The course exists in the studying program of all major universities in Romania and abroad.
- The content of the course is considered by the software companies as important for average programming skills.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	The correctness and completeness of the accumulated knowledge and the capacity to design and implement correct	Written exam.	40%
10.5 Seminar/laboratory	Be able to design and implement a Python program	Practical Exam	30%
	Correctness of assignments and in-lab practical work	Assignments and/or in-lab practical work	30%
10.6 Minimum standard of performance			
<ul style="list-style-type: none"> • The score is computed from the grades of the Written exam, Pratical Exam and Laboratory grade (proportions 40%, 30% and 30%, respectively). The minimum passing score is 5. <ul style="list-style-type: none"> o minum passing score for written exam is 5. o minimum passing score for practical exam is 5. 			

11. Labels ODD (Sustainable Development Goals)²

	General label for Sustainable Development						

² Keep only the labels that, according to the [Procedure for applying ODD labels in the academic process](#), suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable.*”.

--	--	--	--	--	--	--	--	--

Date:
31.10.2025

Signature of course coordinator

Signature of seminar coordinator

.....

.....

Date of approval:
...

Signature of the head of department

.....