

SYLLABUS

1. Information about the study programme

1.1 Institution of higher education	West University of Timisoara
1.2 Faculty	Chemistry, Biology, Geography
1.3 Department of	Geography
1.4 Field of study	Geography
1.5 Study cycle	Master's degree
1.6 Study programme	Geographic Information Systems

2. Information about the subject/discipline

2.1 Name	Introduction to programming						
2.2 Course coordinator	Conf. Dr. Dornik Andrei						
2.3 Seminar coordinator	Conf. Dr. Dornik Andrei						
2.4 Year of study	1	2.5 Semester	1	2.6 Type of assessment	E ¹	2.7 Type of discipline	DS/DOB

3. Total estimated time (hours of teaching per semester)²

3.1 Number of hours per week	3	3.2 course	1	3.3 seminar/laboratory	2
3.4 Total hours in the curriculum	42	3.5 course	14	3.6 seminar/laboratory	28
Distribution of time:					hours
Study based on Instructions, course materials, bibliography and notes					35
Additional documentation library, specialized electronic platforms / field					35
Training seminars / laboratories, homework, essays, portfolios and essays					20
Tutoring					10
Examinations ³					8
Other activities					
3.7 Total hours of individual study	100				
3.8 Total hours per semester⁴	150				
3.9 Number of credits	6				

¹ According to article 37, paragraph (1) of the Higher Education Law no. 199/2023, with subsequent amendments and additions, "the academic success of a student during a study program is determined by verifying the acquisition of the expected learning outcomes through exam-type evaluations and evaluation throughout the semester".

² The total number of contact hours and individual study hours will be aligned with the number of credits allocated to the course. One credit corresponds to a total between 25 and 30 hours of teaching activities and individual study. At the level of academic departments may establish, by discipline categories, the exact equivalence between one credit and the number of hours.

³ The hours corresponding to examinations are added only to the point 3.8 – The total hours per semester, not to be added to the point 3.7 – Total hours of individual study.

⁴ Total hours per semester = total hours in the curriculum + total hours of individual study + hours allocated to examinations.

4. Prerequisites (where applicable)

4.1 of curriculum	Basics in informatics; Geographic Information Systems; Geoinformatics
4.2 of skills	Basic skills of computer use; analytical spirit and the ability to break down problems into sub-problems

5. Conditions (where applicable)

5.1 for the course	• Computer / laptop with audio-video system for the teacher and students • internet access; access to the Elearning UVT platform; video projector
5.2 for the seminar	• complete fulfilment of tasks of laboratory work and projects • Computer / laptop with audio-video system for the teacher and students; • internet access; access to the Elearning UVT platform; video projector

6. Discipline objectives - expected learning outcomes which contribute to the completion and passing the discipline

Knowledge	• Basic knowledge of computer science and mathematics • Concepts related to the structure and operation of a computing system • Concepts and methodologies regarding the analysis, design and implementation of computer applications • Understanding the operation of an algorithm
Skills	• Create algorithms in pseudocode • Analysis of the complexity and correctness of a code • Implementation and testing of a program • The ability to identify algorithms and data structures appropriate to a particular problem, to apply the principles of computer application development, and to implement algorithms in a programming language • The ability to use programming environments/tools/platforms specific to each stage in the development of program • The ability to use file systems, to manage processes specific to a computing system, to ensure effective communication between software components
Responsibility and autonomy	• Development of a critical and analytical spirit among students; appreciating the advantages of using algorithmic thinking • The ability to solve specific tasks autonomously • The ability to identify/select appropriate solutions and generate innovative ideas • The ability to correctly/effectively identify and plan tasks specific to a particular project • The application of effective and responsible work strategies, based on the principles, norms and values of the code of professional ethics • Application of effective work techniques in a multidisciplinary team, ethical attitude, respect for diversity and multiculturalism, acceptance of diversity of opinion • Self-assessment of the need for continuous professional training for the purpose of insertion and adaptability to the requirements of the labor market • Capitalizing on the results obtained to analyses, studies and geographical projects

7. Contents

The platform through which the course materials in electronic format and other learning/bibliographic resources can be accessed: <https://elearning.e-uvt.ro/>

7.1 Course	Teaching methods	Comments
1. Introduction to algorithms. The notion of algorithm. The objectives of programming. Properties of algorithms. Data and data classifications. Simple processing. Structured processing (sequential, decision, cycle).	Lecture, Interactive presentations, heuristic conversation, problematization and hands-on examples	2 hours
2. Description of algorithms and variables. Pseudocode. Description of fundamental processing and structured data. Examples of simple algorithms (calculations of sums and finite products, approximation of infinite sums, operations on whole numbers, operations on tables). Successive refinement technique and decomposition of an algorithm into subalgorithms.		2 hours
3. Lists and dictionaries. Definition and access to elements. Simple operations on lists. Lists. Elementary sorting methods. The problem and method of inserting, selecting and exchanging neighboring elements (for each method: variants of the algorithm, correctness verification, complexity analysis).		3 hours
4. Functions. Local variables. Parameter specification. Calling functions. Returning the results.		2 hours
5. Working with files. Exception handling.		1 hours
6. Implementation of recursive functions. Implementation of recursive algorithms (generation of permutations, generation of subsets).		2 hours
7. Algorithm debugging. The stages of verifying the correctness of the algorithms. Elements of formal analysis of correctness: preconditions, postconditions, invariants, termination functions. Analysis of the complexity of algorithms. Purpose of the analysis. Analyzed resources. Estimation of execution time (best case, worst case, average case). Examples: finite sums, product of two matrices, minimum determination, sequential search.		2 hours
Bibliography: ➤ Elkner J., Downey A.B., Meyers C., How to think like a computer scientist. Learning with Python,		

Green Tea Press, 2002 ➤ Lutz, M., Learning Python, 3rd edition, O. Reilly, 2007 ➤ Swaroop C. H., Zimmerhoff J., A Byte of Python, 2017, ISBN 1977878490 ➤ Tanimoto S., Introduction to Python for Artificial Intelligence, IEEE Computer Society ➤ Additional references and course presentations are posted on Elearning UVT Platform (https://elearning.e-uvt.ro/)		
7.2. Seminar	Teaching methods	Comments
1. Introduction to Python. Installation. Familiarization with the interface. The read-evaluate-print cycle. Evaluation of expressions. Simple mathematical operations.	Hands-on exercises, case studies, scientific explanation and demonstration.	4 hours
2. Specifying variables (numeric, logical, character strings). Rules for constructing expressions. Explicit display of results. Specifying conditional and repetitive processing.		2 hours
3. Lists and dictionaries. Definition and access to elements. Simple operations on lists.		4 hours
4. Definition of functions. Local variables. Parameter specification. Calling functions. Returning the results.		4 hours
5. Working with files. Exception handling.		2 hours
6. Processing of lists. Implementation of search and sorting algorithms		2 hours
7. Implementation of recursive functions. Implementation of recursive algorithms (generation of permutations, generation of subsets).		2 hours
8. Other data types (tuples). The difference between modifiable and non-modifiable types. Implementation of heuristic algorithms.		2 hours
9. Peculiarities of working with multidimensional lists. Implementation of optimization algorithms based on dynamic programming.		2 hours
10. Modules and packages. Creation and use. Import and reload functions. Namespaces.		2 hours
11. Evaluation, Feedback		2 hours
Bibliography: ➤ Downey A.B., How to think like a computer scientist. Learning with Python, Green Tea Press, 2002 ➤ Lutz, M., Learning Python, 3rd edition, O. Reilly, 2007 ➤ Swaroop C. H., Zimmerhoff J., A Byte of Python, 2017, ISBN 1977878490 ➤ Tanimoto S., Introduction to Python for Artificial Intelligence, IEEE Computer Society ➤ Additional references and course presentations are posted on Elearning UVT Platform (https://elearning.e-uvt.ro/)		

8. Corroboration of the course contents with the epistemic expectations of the community representative, professional associations and representative employers of the programme itself

The content of the discipline was developed in accordance with the curriculum and meets the didactic and scientific requirements corresponding to similar specializations in other university centers. Introduction to programming facilitates the acquisition of basic knowledge in carrying out a research project, both from a theoretical point of view and from the point of view of working methods in the field, developing students' analytical thinking, the ability to problematize, to manage a scientific approach, of a database and its operation. The software used in the practical applications are among the most modern and frequently used in specialized institutions. Such applied training makes students compatible with the job market in the field of geographic information systems, or research activity.

9. Use of tools based on generative artificial intelligence

To complete the tasks defined in the assessment section, only for laboratory, the use of generative AI tools is permitted for generating ideas/slogans/designs/images/text rewriting, editing/reviewing. For the Course assessment section, the use of generative AI tools is not permitted. The most well-known examples of generative AI tools include, but are not limited to: ChatGPT, Google Gemini, Copilot for text, or MidJourney for images.

Each student will specify, in a statement written separately for each assignment, according to the model in Annex 3 of the [Regulation on the use of generative artificial intelligence in the educational process at UVT](#), the tool they used, how it was used, and the part of the assignment in which it was used. The statement will be included by the student at the beginning of the submitted assignment.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of the final mark
10.4 Course	Understanding and assimilation of theoretical knowledge	Test	20%
10.5 Seminar	Develop a program in Python	Continuous formative evaluation – presentation and feedback during the semester	20%
		Presentation of the final Python program/code	40%
	Test programming skills	Practical test	20%
10.6 Minimum performance standards			
Minimum mark 5 at course evaluation and minimum mark 5 at practical activities.			

Date of submission:

14.09.2026

Date of approval in department:

Titular of the course: Conf. Dr. Andrei Dornik

Signature:

Seminary titular: Conf. Dr. Andrei Dornik

Signature:

HEAD OF THE DEPARTMENT: Conf. Dr. Ioan Sebastian Jucu