SYLLABUS

PROGRAMMING BASICS

2025-2026

1. Program General Data

1.1. University	**„1 Decembrie 1918” University of Alba Iulia**
1.2. Faculty	**Faculty of Informatics and Engineering**
1.3. Department	**Informatics, Mathematics and Electronics Department**
1.4. Area	**Computer Science**
1.5. Level	**Bachelor**
1.6. Specialization	**Computer Science** **ESCO-08: 2511/ Systems Analyst, 2512/ Software developers Analyst 251201** **Computer System Programmer 251204** **Computer System Engineer 251203**

2. Subject General Data

2.1. **Subject**	**Programming basics**			2.2. **Code**		CSE103	
2.3. **Course holder/ Lecturer/**			**Lect. univ. dr. Nagy – Onița Daniela Marcela**				
2.4. Teaching Assistant's Name			**Asist. univ. drd. Cristea Daniela**				
2.5. Year	**I**	2.6. Semester	**I**	2.7. Evaluation form (E – final exam/C-examination /VP)	**E**	2.8. Status (**C**– Compulsory, **Op** – optional, **F -** Facultative)	**C**

3. Course Structure (Weekly number of hours)

3.1. Weekly number of hours	**4**	3.2. course	***2***	3.3. seminar, laboratory	***2***
3.4. Total number of hours according to the curricula	**56**	3.5. course	***28***	3.6. seminar, laboratory	***28***
Time distribution:					Hours
Individual study using the lecture notes					**45**
Documentation (library)					**22**
Homework, Essays, Portfolios					**56**
Tutoring					
Evaluation (exams)					***2***
Other activities…….					

3.7 Total number of hours for individual study	**69**
3.8 Total number of hours according to the curricula	**56**
3.9 Total number of hours per semester	**125**
3.10 Credits	**4**

4. Prerequisites

4.1. Curricula prerequisites	-
4.2. according to the general competencies	-

5. Conditions

5.1. Conditions to support teaching	*Room equipped with video projector/board.*
5.2. Conditions for supporting seminar/laboratory activities	*Laboratory – computers.*

6. Competențe specifice acumulate (cele alese de titular din grila de competente)

Professional competences	**CP10 (1 ECTS)** – fixes software errors **CP24 (1 ECTS)** – uses user-driven design methodologies **CP27 (1 ECTS)** – uses software libraries **CP29 (1 ECTS)** – uses software design patterns **CP33 (1 ECTS)** – uses computer-aided software engineering tools
Transversal competences	**-**

6.2 Specific learning outcomes for the field and branch of science as provided in the current ARACIS standards

Knowledge	• The student/graduate identifies, explains, and argues fundamental concepts of data structures, algorithms, and programming paradigms, as well as computer architecture.
Skills	• The student/graduate designs, develops, and demonstrates complex software solutions using efficient algorithms and diverse programming paradigms.
Responsibilities and autonomy	• The student/graduate coordinates technical teams for the development of software applications, assuming responsible decisions regarding their optimization and integration.

7. Course objectives

6.1 General course objectives	*Acquisition of basic knowledge and skills in computer programming using the Python language, aimed at algorithmic problem-solving and the development of logical and analytical thinking.*
6.2 Specific course objectives	*Development of basic competencies in the use of the Python language through the understanding of fundamental programming concepts, the construction of mathematical models, the definition and use of functions, as well as the algorithmic solution of problems. Students will learn to write clear and correct code, debug programs, and apply introductory programming concepts, thereby developing logical thinking and analytical skills. This includes the creation of applications and the solving of problems using programming language instructions.*

8. Course contents

8.1 Curs	Teaching methods	References
1. • Introductory concepts of programming and Python (2 hours)	Lecture, discussions, problem analysis, and examples	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
2. • Data types and variables. Operators and expressions. Input/Output statements (2 hours)	Lecture, discussions, problem analysis, and examples	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
3. • Control structures (2 hours)	Lecture, discussions, problem analysis, and examples	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
4. • Iterative structures (2 hours)	Lecture, discussions, problem analysis, and examples	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
5. • Functions (2 hours)	Lecture, discussions, problem analysis, and examples	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
6. • Lists and list operations. Tuples, sets, and dictionaries (2 hours)	Lecture, discussions, problem analysis, and examples	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python

		Programming. OpenStax, 2024.
7. • Regular expressions in Python (2 hours)	Lecture, discussions, problem analysis, and examples	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
8. • Standard Python libraries: *time, random, hashlib*. Libraries for data serialization in Python (*json / xml*). Working with zip archives (2 hours)	Lecture, discussions, problem analysis, and examples	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
9. • Strings and string-specific operations. Composite and nested data structures (2 hours)	Lecture, discussions, problem analysis, and examples	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
10. • Error and exception handling (2 hours)	Lecture, discussions, problem analysis, and examples	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
11. • Basic algorithms (4 hours)	Lecture, discussions, problem analysis, and examples	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
12. • Practical applications and simple projects (4 hours)	Lecture, discussions, problem analysis, and examples	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.

References

Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022.
Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
Python Basics: A Practical Introduction to Python 3, David Amos, Dan Bader, Joanna Jablonski, Fletcher Heisler, 2020

Introduction to Python Programming, UDAYAN DAS, SAINT MARY'S COLLEGE OF CALIFORNIA AUBREY LAWSON, WILEY CHRIS MAYFIELD, JAMES MADISON UNIVERSITY NARGES NOROUZI, UC BERKELEY, 2024 Learning Python, Mark Lutz Python Essential Reference, David Beazley Python: The Fundamentals Of Python Programming, Paul Jones, 2016

8.2. Laboratory	**Teaching methods**	**References**
1. • Introduction to the Python language. Installation. How to run a Python program? (2 hours)	Examples, exercises, practical applications	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
2. • Variable declarations and data types. Use of operators and expression evaluation. (2 hours)	Examples, exercises, practical applications	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
3. • Data input and output. Use of conditional structures. (2 hours)	Examples, exercises, practical applications	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
4. • Use of iterative structures (2 hours)	Examples, exercises, practical applications	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
5. • Defining and using functions (2 hours)	Examples, exercises, practical applications	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
6. • Nested data structures. Use of lists and tuples. Problem-solving. (2 hours)	Examples, exercises, practical applications	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
7. • Use of regular expressions. Problem-solving. (2 hours)	Examples, exercises, practical applications	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python

		Programming. OpenStax, 2024.
8. • Serialization. Standard Python libraries. Problem-solving. (2 hours)	Examples, exercises, practical applications	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
9. • Strings – processing. Sets and dictionaries (hash tables) in Python. Concepts of functional programming. (2 hours)	Examples, exercises, practical applications	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
10. • The Python exception system. I/O operations. Python modules – *sys* and *os*. (2 hours)	Examples, exercises, practical applications	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
11. • Algorithmic problem-solving. Implementation of simple algorithms: search, max/min selection, sorting, etc. (4 hours)	Examples, exercises, practical applications	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.
12. • Practical project, individual or team-based (4 hours)	Examples, exercises, practical applications	1. Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022. 2. Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.

References

Adriana STAN, Introducere în Python folosind Google Colab, UTPress, 2022.

Udayan, Das, et al. Introduction to Python Programming. OpenStax, 2024.

Python Basics: A Practical Introduction to Python 3, David Amos, Dan Bader, Joanna Jablonski, Fletcher Heisler, 2020

Introduction to Python Programming, UDAYAN DAS, SAINT MARY'S COLLEGE OF CALIFORNIA AUBREY LAWSON, WILEY CHRIS MAYFIELD, JAMES MADISON UNIVERSITY NARGES NOROUZI, UC BERKELEY, 2024

Learning Python, Mark Lutz

Python Essential Reference, David Beazley

Python: The Fundamentals Of Python Programming, Paul Jones, 2016

Resurse web:

https://docs.python.org/3/

9. Corroborating Course content expectations to the epistemic community representatives, professional associations and employers representative for the curricula

Following periodic consultations with representatives of the epistemic community, professional associations, and relevant employers in the field of information technology, the content of the course *Introduction to Programming (Python)* has been validated and positively evaluated in its current form. These consultations confirmed the relevance of the topics covered for the development of essential competencies in the field, emphasizing algorithmic thinking, familiarity with modern programming languages, and the development of practical skills demanded by the labor market.

10. **Assessment**

Activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage from the final mark
10.4 Course	*Final evaluation*	*Written exam*	50 %
	-	-	-
10.5 Seminar/laboratory	*Continuous assessment*	*Portfolio of laboratory*	50 %
10.6 Minimum performance standard:			
Achieving a minimum grade of 5 in both the final assessment and the evaluation of the portfolio of assignments and the final project.			
Demonstration of competencies in: The student must demonstrate an understanding of fundamental programming concepts and be able to develop simple Python programs using variables, control structures, functions, and basic data structures (lists, strings). Additionally, the student should be able to interpret simple requirements, design basic algorithms, and write code that is syntactically and logically correct, with a minimal degree of autonomy.			

Completion date	Instructor's signature Lect. univ. dr. Nagy-Onița Daniela	Teaching assistant's signature Asist. univ. drd. Cristea Daniela

Date of approval by Department members

Department director signature
Lect.univ.dr. Aldea Mihaela

Date of approval by Faculty Council

Signature of the Dean
Conf.Univ.dr. Rotar Corina