

FUNDAMENTALS OF COMPUTER PROGRAMMING

Course code	<i>IT111</i>
Compulsory in the programmes	<i>Economics and Data Analytics</i>
Level of studies	<i>Undergraduate</i>
Number of credits	<i>6 ECTS (42 in-class hours + 6 hours of examination, 114 individual work hours)</i>
Course coordinator (title and name)	<i>Kristina Aldošina</i>
Prerequisites	-
Language of instruction	<i>English</i>

THE AIM OF THE COURSE

The aim of this course is to provide students with a solid foundation in Python programming, equipping them with the essential concepts, skills, and confidence to write clear, functional, and well-structured code. Through a hands-on, step-by-step approach, learners will be introduced to fundamental programming constructs such as variables, data types, collections, conditional statements, and loops, progressing gradually toward more advanced topics including functions, error handling, and object-oriented programming with classes.

By the end of the course, students will not only be capable of solving basic to intermediate-level programming problems but will also be fully prepared to continue their learning journey into specialized areas such as data analysis and visualization. This course serves as a crucial steppingstone for further studies and applications in Python-driven data science.

MAPPING OF COURSE LEVEL LEARNING OUTCOMES (OBJECTIVES) WITH DEGREE LEVEL LEARNING OBJECTIVES (See Annex I), ASSESSMENT AND TEACHING METHODS

Course level learning outcomes (objectives)	Learning objectives for the Bachelor of Social Science	Assessment methods	Teaching methods
CLO1. To be able to process information and operate at multiple levels of abstraction.	ELO4.1. Students will be able to communicate reasonably in different settings according to target audience tasks and situations	Midterm exam, final exam, project	Lectures, seminars, individual work
CLO2. To be able to decompose IT problems into parts and solve them efficiently.	ELO1.2. Students will be able to identify underlying assumptions and logical consistency of causal statements	Midterm exams, final exam	Lectures, seminars, individual work
CLO3. To be able to demonstrate proficiency in a software development environment.	ELO3.2. Students will be able to make decisions using appropriate IT tools	Midterm exam, final exam, project	Seminars, individual work
CLO4. To be able to assess the correctness, design, and style of code.	ELO3.2. Students will be able to make decisions using appropriate IT tools	Midterm exam, final exam, project	Seminars, individual work

ACADEMIC HONESTY AND INTEGRITY

The ISM University of Management and Economics Code of Ethics, including cheating and plagiarism are fully applicable and will be strictly enforced in the course. Academic dishonesty, and cheating can and will lead to a report to the ISM Committee of Ethics. With regard to remote learning, ISM remind students that they are expected to adhere and maintain the same academic honesty and integrity that they would in a classroom setting.

COURSE OUTLINE

Topic	In-class hours	Readings
Introduction. Computer science, binary numbers, programming languages, Python, PEP8 standard, code editors.	2	Lecture notes. https://www.w3schools.com/python/default.asp (intro, get started, syntax, comments, variables, data types, numbers, casting, strings, booleans, operators, lists, tuples, sets, dictionaries, input, string formatting, Python reference)
1. Variables, integers, floats, booleans.	4	
2. Strings, input, formatted strings.	4	
3. Lists, tuples.	4	
4. Dictionaries, sets.	4	
MIDTERM EXAM	2	
5. Conditional statements, "while" loop.	4	Lecture notes. https://www.w3schools.com/python/default.asp (if...else, while loops, for loops, functions, lambda, modules, dates, math, try...except, polymorphism, iterators, scope, Python reference, module reference)
6. "for" loop, comprehensions.	4	
7. Built-in and user-defined functions.	4	
8. Lambda functions, error handling.	4	
9. Modules, import.	4	
10. Introduction to object-oriented programming, classes.	4	Lecture notes. https://www.w3schools.com/python/default.asp (classes/objects, inheritance)
FINAL EXAM	4	
Total:	48	
GROUP PROJECT	2	

FINAL GRADE COMPOSITION

Type of assignment	%
<i>Individual Components 70%</i>	
Midterm exam (topics 1 – 4)	30
Final exam (topics 5 – 10)	40
<i>Group Components 30%</i>	
Group project	30
Total:	100

DESCRIPTION AND GRADING CRITERIA OF EACH ASSIGNMENT

The overall assessment of the course (total maximum of 100% is possible) will be composed from evaluations of multiple tasks which are described as follows:

1. **Midterm exam** is 90 minutes long in-class open-book individual examination. Students will have to solve one complex coding task. Detailed rules (how to complete, how to ensure academic honesty, where to upload, evaluation criteria, etc.) will be announced on e-learning.

2. **Final exam** is 180 minutes long in-class open-book individual examination. Students will have to solve one complex coding task. Detailed rules (how to complete, how to ensure academic honesty, where to upload, evaluation criteria, etc.) will be announced on e-learning.
3. **Group project** is both technical and creative task. For this project, a group of students will have to choose one Python module (which is not related to data analysis, data science, deep learning or any other area which will be covered in later courses), to build a program around it, to present this program to the lecturer and groupmates, and to defend it (answer questions). Detailed rules and requirements for this assignment will be announced on e-learning.

RETAKE POLICY

In case of a negative final evaluation, retake is possible. It will cover material of the whole course and will comprise **70%** of the final grade. Marks earned for midterm exam, and final exam will be annulled; mark earned for group project will stay intact. Retake is 180 minutes long in-class open-book individual examination. Students will have to solve one complex coding task. Detailed rules (how to complete, how to ensure academic honesty, where to upload, evaluation criteria, etc.) will be announced on e-learning.

COURSE SPECIFIC AI GUIDELINES

All course participants must comply with the ISM University of Management and Economics Artificial Intelligence Policy published on the e-learning platform.

Group project:

Students may use AI tools as a learning, programming, debugging, and documentation support aid. AI tools may be used to clarify Python concepts, understand documentation of Python modules, brainstorm project ideas, suggest program structure, generate code, debug errors, refactor code, improve code readability, prepare test cases, and improve the clarity of written explanations or presentation text.

For the group project, fully or partially AI-generated code is allowed. However, students remain fully responsible for the final submitted program. Each group member must understand the overall structure of the program, the purpose of every file or module, and every command, function, class, method, library, and parameter used in the submitted code. During the project presentation and defense, students must be able to explain how the program works, why specific programming decisions were made, and how AI-generated suggestions were verified, modified, or integrated.

AI-generated code or explanations must not be submitted blindly. If students cannot explain the code, the logic, or the role of specific commands and functions, this may negatively affect the project evaluation. Any substantial use of AI tools must be acknowledged according to the ISM AI Policy and assignment instructions. AI tools must not be used to fabricate sources, examples, data, or functionality.

Seminar and independent learning tasks:

Students may use AI tools to ask for clarification, request additional examples, practice similar coding problems, receive hints, compare alternative solutions, or get feedback on errors after making their own attempt. AI should support learning and understanding, not replace it. Unless explicitly allowed by the course instructor, AI tools are not permitted during in-class assessments, including the midterm exam, final exam, and retake.

ADDITIONAL REMARKS

Precision of composite evaluations is left intact (up to 2 decimal places) until the end of semester and only the final evaluation will be subject to rounding.

REQUIRED READINGS

1. Lecture notes.
2. <https://www.w3schools.com>

ADDITIONAL READINGS

1. Effective Python: 59 specific ways to write better Python, Brett Slatkin, 2015, ISBN 978-0-13-403428-7
2. Learning Python, Fifth edition, Mark Lutz, 2013, ISBN: 978-1-449-35573-9
3. Python Cookbook, Third Edition, David Beazley and Brian K. Jones, 2013, ISBN: 978-1-449-34037-7

4. Python Crash Course, Eric Matthes, 2016, ISBN-10: 1-59327-603-6

MORE EXERCISES

1. <https://www.hackerrank.com/domains/python>
2. <https://edabit.com/challenges>
3. <https://holypython.com/beginner-python-exercises/>
4. <https://www.hackerrank.com/domains/python>
5. <https://codesignal.com/>
6. <https://pythontutor.com/index.html>

DEGREE LEVEL LEARNING OBJECTIVES

Learning objectives for the Bachelor of Social Science

Programmes:

Economics and Data Analytics,

Economics and Politics

Learning Goals	Learning Objectives
Students will be critical thinkers	ELO1.1. Students will be able to understand core concepts and methods in the key economics disciplines
	ELO1.2. Students will be able to identify underlying assumptions and logical consistency of causal statements
Students will have skills to employ economic thought for the common good	ELO2.1. Students will have a keen sense of ethical criteria for practical problem-solving
Students will be technology agile	ELO3.1. Students will demonstrate proficiency in common business software packages
	ELO3.2. Students will be able to make decisions using appropriate IT tools
Students will be effective communicators	ELO4.1. Students will be able to communicate reasonably in different settings according to target audience tasks and situations
	ELO4.2. Students will be able to convey their ideas effectively through an oral presentation
	ELO4.3. Students will be able to convey their ideas effectively in a written paper