COURSE DESCRIPTION CARD - SYLLABUS

Course name
Introduction to Programming in Python [S1DSwB1>WdPP]

Course

Field of study
Data Science in Business

Year/Semester
1/1

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
30

Laboratory classes
0

Other
0

Tutorials
30

Projects/seminars
0

Number of credit points

5,00

Coordinators

dr inż. Marcin Nowak
marcin.nowak@put.poznan.pl

dr Grzegorz Nowak
grzegorz.nowak@put.poznan.pl

Lecturers

Prerequisites

The student has basic knowledge of mathematics and computer science. They are able to learn effectively and solve problems analytically. They are aware of the necessity of expanding their competencies.

Course objective

The aim of this course is to introduce students to the fundamental principles of programming in Python. Students will acquire knowledge of the basic syntax and semantics of the language, methods of code organization, and key concepts such as data types, operators, control statements, functions, and data structures. The course aims to develop analytical thinking skills and the ability to effectively solve algorithmic problems. Students will learn to write, test, and debug Python programs, as well as utilize basic libraries and development tools. The acquired competencies will provide a solid foundation for further development of programming skills in more advanced areas of computer science.

Course-related learning outcomes

Knowledge:
Describes the basic concepts of programming, Python syntax, and its key data structures [DSB1_W02].
Explains the fundamental mechanisms of debugging, code testing, and API handling in Python [DSB1_W05].

Skills:
Implements programs in Python, using variables, operators, data structures, and control statements [DSB1_U02].
Creates and uses functions and modules to organize code in Python [DSB1_U08].
Processes text files, JSON, XML, and other data formats using Python [DSB1_U04].
Uses Pandas and NumPy libraries for data analysis and processing [DSB1_U09].
Debugs and tests Python code, utilizing error analysis tools to ensure program correctness [DSB1_U07].
Argues and evaluates different approaches to solving programming problems based on code analysis [DSB1_U11].
Independently develops programming skills by utilizing available resources and online courses [DSB1_U15].

Social competences:
Critically analyzes code correctness, optimizing its performance and adherence to best programming practices [DSB1_K01].
Collaborates in programming teams, using version control tools and collaborative code editing platforms [DSB1_K02].
Applies automation technologies and Python scripts to improve work efficiency and optimize business processes [DSB1_K04].

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lecture:
There are two tests, each graded in the form of points-50 points per test. The final grade is determined by the sum of points from both tests. The first test takes place midway through the course, while the second is held at the end. The passing threshold is a total of 50 points from both tests.
Laboratories:
There are two tests, each graded in the form of points-50 points per test. The final grade is determined by the sum of points from both tests. The first test takes place midway through the course, while the second is held at the end. The passing threshold is a total of 50 points from both tests.

Programme content

The course covers fundamental programming concepts and an introduction to the Python language, including its syntax, data types, and operators. Key data structures and control statements, such as loops and conditional statements, are discussed. Participants will learn the principles of modular and functional programming and how to work with files and input-output streams. The course also includes an introduction to Python libraries and API handling. Additionally, topics such as debugging, code testing, and task automation using Python scripts are covered.

Course topics

Fundamental concepts of programming
Introduction to Python and the development environment
Basics of Python syntax - variables, operators, data types
Data structures - lists, tuples, dictionaries, sets
Control statements - loops and conditionals
Functions and modular programming
File handling and input-output streams
Functional programming - lambda, map, filter, reduce
Advanced file handling - JSON, XML, pickle
Introduction to Python libraries
Pandas and NumPy as key data analysis libraries
API handling and data retrieval from the internet

Multithreading programming
Debugging and code testing
Task automation and Python scripts

Teaching methods

Lectures: Problem-based lecture, case study presentation
Laboratories: Problem-solving tasks, case study analysis, group work

Bibliography

Basic:
Mertz, D. (2024). Lepszy kod w Pythonie, Promise
Matthes, E. (2023). Python. Instrukcje dla programisty, Helion, Gliwice

Additional:
Dawson, M. (2023). Python dla każdego. Podstawy programowania, Wydanie 3, Helion, Gliwice.
Downey, A.B. (2025). Myśl w języku Python! Nauka programowania, Helion

Breakdown of average student's workload

	Hours	ECTS
Total workload	125	5,00
Classes requiring direct contact with the teacher	62	2,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	63	2,50