Module/Course Syllabus
Program: COMPUTER SCIENCE
Full-time master degree program

Course:	**Advanced programming in Python**
Type of the course:	elective
Course code:	I2S2.13
Year:	I
Semester:	2
Form of the degree program:	full-time
Form of classes and number of hours per semester:	60
Lecture	30
Classes	0
Laboratory	30
Project	0
Number of ECTS credits:	4
Form of assessment:	course completion assessment
Course language:	English

Course objective (CO)	
CO1	Familiarizing students with advanced methods of programming in Python
CO2	Acquisition by students the ability to work with libraries that extend the functionalieties of the Python language.

Prerequisites in terms of knowledge, skills and other competencies	
1	Basic programming in algorithmic languages.
2	Basic knowledge of Python.
3	Knowledge of the basics of algorithms and data structures.

Learning outcomes (LO)	
	In terms of knowledge:
LO 1	He has knowledge of advanced programming in Python.
LO 2	He has knowledge of the most important programming paradigms in Python.
	In terms of skills:
LO 3	He has ability to functional and object-oriented programming in Python.
LO 4	He can create multi-threaded applications in Python.
	In terms of social competence:
LO 5	He is ready to take on challenges related to the use of modern programming techniques.
LO 6	Is ready to cooperate with colleagues and the teacher in laboratory classes.

Course content	
Form of classes – lectures (L)	
	Course content
L1	Organization of the working environment. Isolated virtualenv environments, library management, adding and removing libraries from the environment.
L2	Advanced Typing in Python. Use of Type annotation and typing module. IDE support for variable type analysis.

L3	Complex types and their extensions. Collections module.
L4	Functional programming I. Lambda functions, Special arguments args and kwargs.
L5	Functional programming II. A function as a variable, a function argument, and functions that return functions.
L6	Object-oriented programming I. Classes and instances, class properties and methods, static methods. Documenting the code.
L7	Object-oriented programming II. Creating properties using decorators, dynamically adding methods to a class. Inheritance.
L8	Object-oriented programming III. Polymorphism, operator overloading and abstract classes. Exceptions.
L9	Application testing. Unit tests for the created code.
L10	Regular expressions. Working with the re module, defining regular expressions and finding them.
L11	Multithreaded programming I. Introduction to multithreading. Creating and managing threads.
L12	Multithreaded programming II. Locks, semaphores and decision variables.
L13	Asynchronous Python. Working with the AsyncIO module.
L14	Support for data in JSON and XML format. JSON in Python. DOM in Python, navigating an XML document.
L15	Database integration. Working with MySQL and MongoDB. Object-Relational Mapping (ORM).
Form of classes – laboratories (Lab)	
	Course content
Lab1	Isolated virtualenv environment. Environment creation, overview of installed libraries, adding and removing libraries. Moving the environment.
Lab2	Typing in Python. Use of Type annotation and typing module in practical examples. Creating packages in Python and the use of the __init__.py file.
Lab3	Complex types and their extensions. Work with built-in complex types and with the Collections module. Applications of the NamedTuple, DataClass, DefaultDict, Deque, and Counter types.
Lab4	Aspects of the functional programming. Classic, anonymous and lambda functions. Use of special arguments args and kwargs. Using a function as a variable, a function argument, and functions that return functions.
Lab5	Object-oriented programming. Creating a class and instance, class properties and methods, static methods. Documentation of the created code. Organizing classes into packages. Creating properties using decorators, dynamically adding methods to a class. Inheritance, polymorphism, operator overloading and abstract classes. Exceptions.
Lab6	Application testing. Unit tests for the created code.
Lab7	Regular expressions. Working with the re module, defining regular expressions, regular expression syntax, match and search functions. Using flags.
Lab8	Multithreaded programming. Multithreading implementation. Creating and managing threads. Locks, semaphores and decision variables.
Lab9	Asynchronous Python. Working with the AsyncIO module.
Lab10	Support for data in JSON and XML format. JSON in Python, converting Python objects to JSON. DOM in Python, navigating an XML document, and creating a new XML document.
Lab11	Database integration. Working with MySQL and MongoDB. Object-Relational Mapping (ORM).

Didactic methods	
1	Informative lecture with multimedia presentation.

2	Laboratory exercises: implementation of tasks using the learned programming methods.
3	Thematic discussion.

Assessment methods and criteria		
Assessment method symbol	**Assessment method description**	**Passing threshold**
A1	Lecture: written exam (multiple-choice test and open questions) from the curriculum content (questions concern each of the areas W1, ..., W15 with equal weight in the final grade).	**51%**
A2	Laboratory: evaluation of the results of the work - the results of laboratories L1 - L11.	**51%**

Required textbooks and other course materials	
1	Ramalho L., Zaawansowany python. Jasne, zwięzłe i efektywne programowanie, APN Promise 2015
2	Lee K. D., Python Programming Fundamentals, Springer 2014
3	Romano F., Learning Python, Packt 2015
Recommended textbooks and other course materials	
1	Vinci S., Python Programming Cookbook, Exelixis Media P.C. 2016
2	Matthes E., Python. Instrukcje dla programisty, Helion 2020
3	Stephenson B., The Python Workbook A Brief Introduction with Exercises and Solutions, Springer 2014

Student workload	
Form of activity	**Average number of hours to complete the activity**
Contact hours with the lecturer, including:	60
participation in lectures	30
participation in laboratories	30
Student's own work, including:	40
preparation for the exam	20
preparation for the laboratory	20
Total student workload	100
Total number of ECTS credits	4

Learning outcomes matrix					
Learning outcome	**Reference to learning outcomes defined for the master's program**	**Course objectives**	**Course content**	**Didactic methods**	**Assessment methods**
LO 1	**I2A_W07 +++**	*CO1*	*L1 - L15*	*1*	*A1*
LO 2	**I2A_W01 ++ I2A_W08 +**	*CO1*	*L1 - L15*	*1*	*A1*
LO 3	**I2A_U07 ++ I2A_U13 +++**	*CO2*	*Lab1 - Lab11*	*2, 3*	*A2*
LO 4	**I2A_U01 ++ I2A_U15 ++**	*CO2*	*Lab1 - Lab11*	*2, 3*	*A2*
LO 5	**I2A_K02 ++**	*CO2*	*Lab1 - Lab11*	*2, 3*	*A2*
LO 6	**I2A_K05 ++**	*CO2*	*Lab1 - Lab11*	*2, 3*	*A2*

The author of the program:	Michał Dolecki, PhD
E-mail address:	m.dolecki@pollub.pl
Organizational unit:	Department of Computer Science