Kielce University of Technology
FACULTY OF MANAGEMENT AND COMPUTER MODELLING

MODULE DESCRIPTION

Module code	full-time studies:	**Z-ZIP1-E-508b**
	part-time studies:	**Z-ZIPN1-E-508b**
Module name	**Programming Languages - Python**	
Module name in Polish	**Języki programowania - Python**	
Valid from academic year	**2019/2020**	

MODULE PLACEMENT IN THE SYLLABUS

Field of study	**MANAGEMENT AND PRODUCTION ENGINEERING**
Level of education	**1st degree**
Studies profile	**General**
Form and method of conducting classes	**Full-time and Part-time**
Specialisation	**All**
Unit conducting the module	**Department of Computer Science Technologies**
Module co-ordinator	**Paweł Stąpór, PhD**
Approved by:	

MODULE OVERVIEW

Type of subject / group of subjects	**Major**
Module status	**Non-compulsory**
Language of conducting classes	**English**
Module placement in the syllabus - semester	**Semesetr V**
Initial requirements	**Fundamentals of Computer Science**
Examination (YES/NO)	**NO**
Number of ECTS credit points	**3**

Method of conducting classes		Lecture	Classes	Laboratory	Project	Other
Per semester	full-time studies:	**15**		**30**		
	part-time studies:	**9**		**18**		

TEACHING RESULTS AND THE METHODS OF ASSESSING TEACHING RESULTS

Category	Symbol	Learning outcomes	Assignations to the directional learning outcomes
Knowledge	W01	Student ma poszerzoną wiedzę nt. programowania proceduralnego i obiektowego. Rozumie pojęcia dziedziczenia, kompozycji i polimorfizmu obiektów w odniesieniu do języka Python	ZIP1_W05
	W02	Student rozumie i zna zasady budowy aplikacji obsługujących bazy danych.	ZIP1_W04
	W03	Student ma wiedzę nt. zasad projektowania interfejsu GUI aplikacji zgodnego z architekturą systemu operacyjnego	ZIP1_W05
Skills	U01	Student potrafi budować aplikacje w języku Python dla system Windows z wykorzystaniem plików	ZIP1_U07
	U02	Student potrafi ocenić przydatność narzędzi programowania do rozwiązywania zagadnień z zakresu inżynierii produkcji	ZIP1_U01
Social competences	K01	Student rozumie potrzebę stałego uzupełniania wiedzy z obszaru nowoczesnych narzędzi i idei informatyki.	ZIP1_K01

TEACHING CONTENTS

Method of conducting classes	Teaching contents
Lecture	The concept of module and attributes. Ways of executing a Python program, importing and reloading modules, IDLE interface. Types of built-in objects, instructions, aspects of functional programming, special modes of passing function arguments, ranges of names and their use Methods of processing text files, saving Python objects to a file, *pickle* method Object-oriented programming, inheritance hierarchy, *class* tool, searching for object attributes in the class inheritance tree, class methods definition, *self* argument, *__init__, __add__ and __str__* special methods. Class introspection tools: *__dict__, __class__, __name__* attributes. An example of creating an object-oriented program. Storing objects in a *shelve* database, interactive database operation. An example of a *shelve* database console interface. Connecting to SQL databases, using *sqlite3* and *mysql* modules Graphical user interface GUI, *tkinter* module, widget layout - *pack* and *grid* methods, event handling functions - *bind* and *command* methods. An example of the graphical user interface of the *shelve* database. Examples of advanced database applications in the field of data engineering.

Laboratory	Ways of executing a Python program, importing and reloading modules, IDLE interface Processing of typical Python data structures: lists, tuples, dictionaries and sets; control statements: *if, for, while*. Aspects of functional programming: functions, arguments, ranges of names, special modes of matching arguments Processing data contained in text files, saving Python objects in a file: *pickle* method Object-oriented programming. Class definition, *__init__* constructor, *self* attribute. Class introspection tools: *__dict__*, *__class__*, *__name__* attributes. Special attributes: *__add__*, *__str__*, role of inheritance and overloading. Creating and maintaining a *shelve* database, creating a database console interface. Graphical user interface (GUI) - *tkinter* module. Create a graphical user interface (GUI) for a *shelve* database Support for SQL database in Python. Connecting to SQL databases, using *sqlite3* and *mysql* modules. Creating a database and tables. Populating tables with data.

METODS OF ASSESSING TEACHING RESULTS

Symbol	Methods of checking the learning outcomes *(select X)*					
	Oral exam	Written exam	Test	Project	Statement	Other
W01			X			
W02			X			
W03			X			
U01			X			
U02			X			
K01						X

FORM AND CONDITIONS OF PASSING

Form of classes	Form of credit	Passing conditions
Lecture	Credit with grade	Obtaining at least 50% of the test points in the classroom
Laboratory	Credit with grade	Obtaining at least 50% of test points during the class

STUDENT WORKLOAD

		Balance of ECTS points											
No.	**Type of student's activity**	**Student's workload**											**Unit**
		full-time					**part-time**						
1.	Participation in the activities	Lc	C	Lb	P	O	Lc	C	Lb	P	O		h
		15		30			9		18				
2.	Other (consultation, exam)	2		2			2		2				h
3.	**Number of hours of a student's assisted work**	**49**					**31**						h
4.	**Number of ECTS credit points which are allocated for assisted work**	**2,0**					**1,2**						ECTS
5.	**Number of hours of a student's unassisted work**	**26**					**44**						h
6.	**Number of ECTS credit points which a student receives for unassisted work**	**1,0**					**1,8**						ECTS
7.	**Work input connected with practical classes**	**50**					**50**						h
8.	**Number of ECTS credit points which a student receives for practical classes**	**2,0**					**2,0**						ECTS
9.	**Total number of hours of a student's work**	**75**					**75**						h
10.	**Punkty ECTS za moduł** *1 ECTS=25 hours*	**3**											ECTS

LITERATURE

1. Dawson M. (2014), *Python dla każdego. Podstawy programowania*, Wydawnictwo Helion, Gliwice.
2. Lutz M. (2011), *Python. Wprowadzenie*, Wydanie IV, Wydawnictwo Helion, Gliwice.
3. http://pl.python.org
4. https://docs.python.org/3/