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:	**Dariusz Bojczuk, PhD, DSc**

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	The student has extensive knowledge of procedural and object-oriented programming. He understands the concepts of inheritance, composition and object polymorphism in relation to the Python language.	ZIP1_W05
	W02	The student understands and knows the principles of building applications that support databases.	ZIP1_W04
	W03	The student has knowledge about the principles of designing the application GUI interface in accordance with the architecture of the operating system.	ZIP1_W05
Skills	U01	The student is able to build Python applications for Windows using files.	ZIP1_U07
	U02	The student is able to assess the usefulness of programming tools to solve problems in the field of production engineering.	ZIP1_U01
Social competences	K01	The student understands the need to constantly supplement knowledge in the area of modern tools and ideas of computer science.	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

<table>
<tr><th colspan="13">Balance of ECTS points</th></tr>
<tr><td rowspan="3">No.</td><td rowspan="3">Type of student's activity</td><td colspan="10">Student's workload</td><td rowspan="3">Unit</td></tr>
<tr><td colspan="5">full-time</td><td colspan="5">part-time</td></tr>
<tr><td>Lc</td><td>C</td><td>Lb</td><td>P</td><td>O</td><td>Lc</td><td>C</td><td>Lb</td><td>P</td><td>O</td></tr>
<tr><td>1.</td><td>Participation in the activities</td><td>15</td><td></td><td>30</td><td></td><td></td><td>9</td><td></td><td>18</td><td></td><td></td><td>h</td></tr>
<tr><td>2.</td><td>Other (consultation, exam)</td><td>2</td><td></td><td>2</td><td></td><td></td><td>2</td><td></td><td>2</td><td></td><td></td><td>h</td></tr>
<tr><td>3.</td><td>Number of hours of a student's assisted work</td><td colspan="5">49</td><td colspan="5">31</td><td>h</td></tr>
<tr><td>4.</td><td>Number of ECTS credit points which are allocated for assisted work</td><td colspan="5">2,0</td><td colspan="5">1,2</td><td>ECTS</td></tr>
<tr><td>5.</td><td>Number of hours of a student's unassisted work</td><td colspan="5">26</td><td colspan="5">44</td><td>h</td></tr>
<tr><td>6.</td><td>Number of ECTS credit points which a student receives for unassisted work</td><td colspan="5">1,0</td><td colspan="5">1,8</td><td>ECTS</td></tr>
<tr><td>7.</td><td>Work input connected with practical classes</td><td colspan="5">50</td><td colspan="5">50</td><td>h</td></tr>
<tr><td>8.</td><td>Number of ECTS credit points which a student receives for practical classes</td><td colspan="5">2,0</td><td colspan="5">2,0</td><td>ECTS</td></tr>
<tr><td>9.</td><td>Total number of hours of a student's work</td><td colspan="5">75</td><td colspan="5">75</td><td>h</td></tr>
<tr><td>10.</td><td>Punkty ECTS za moduł
1 ECTS=25 hours</td><td colspan="10">3</td><td>ECTS</td></tr>
</table>

LITERATURE

1. Ernesti J., Kaiser P. (2010), *Python. The Comprehensive Guide*, Rheinwerk Publishing, Quincy, MA.
2. Ledger J.L. (2022), *Python Programming For Beginners*, Independently published.
3. Summerfield M. (2010), *Programming in Python 3: A Complete Introduction to the Python Language (Developer's Library)*, 2nd Edition, Addison-Wesley Professional, Upper Saddle River, NJ · Boston.
4. https://www.python.org/doc