SYLLABUS – A COURSE DESCRIPTION

I. General information
1. Course name: Python Programming
2. Course code: PYPRO
3. Course type (compulsory or optional): compulsory
4. Study programme name: Language, Mind, Technology
5. Cycle of studies (1st or 2nd cycle of studies or full master's programme): 2nd cycle of studies
6. Educational profile (general academic profile or practical profile): general academic
7. Year of studies (if relevant): 1MA
8. Type of classes and number of contact hours (e.g. lectures: 15 hours; practical classes: 30 hours): practical classes: 30 hours
9. Number of ECTS credits: 4
10. Name, surname, academic degree/title of the course lecturer/other teaching staff: : Robert Dyzman, M.Sc.Eng., robdyz@ext.amu.edu.pl
11. Language of classes: English
12. Online learning – yes (partly – online / fully – online) / no: no

II. Detailed information
1. Course aim (aims):
   1.1   Provide knowledge and exercises on creating and using functions and modules.
   1.2   Provide knowledge and exercises on regular expressions usage.
   1.3   Provide knowledge and exercises on advanced text analysis tools.
   1.4   Provide knowledge about input - output operations (keyboard, files).
   1.5   Provide knowledge and exercises in obtaining data from the Internet using the created scripts. Creating simple webcrawlers.
   1.6   Provide knowledge and exercises on Python as an object-oriented language. Creating and using your own classes. Class inheritance.
   1.7   Provide knowledge and exercises in creating the Graphic User Interface.
   1.8   Provide knowledge and exercises about functional programming.

2. Pre-requisites in terms of knowledge, skills and social competences (if relevant):

3. Course learning outcomes (EU) in terms of knowledge, skills and social competences and their reference to study programme learning outcomes (EK):

Course learning outcome symbol (EU)	On successful completion of this course, a student will be able to:	Reference to study programme learning outcomes (EK)
PYPRO_01	knows how to create complex programs, using functions and self-created modules.	K_W10
PYPRO_02	can use regular expressions and advanced text analysis tools. Knows how to read text from a file and save to a file.	K_W10, K_U14, K_U18
PYPRO_03	can use self-created scripts to obtain data from the Internet. Creating simple webcrawlers	K_W10, K_U17, K_U18
PYPRO_04	knows how to create objects, use classes and class-inheritance	K_W10, K_U15
PYPRO_05	knows how to create programs using the Graphic User Interface	K_W10, K_U15
PYPRO_06	understands the concept of functional programming and knows how to use it	K_W10, K_U08, K_U18

4. Learning content with reference to course learning outcomes (EU)

Course learning content:	Course learning outcome symbol (EU)
Functions, their types and use. Creating your own modules.	PYPRO_01
Input-output operations. Regular expressions and text manipulation	PYPRO_02
Internet Data, retrieving Webpages. Webcrawlers	PYPRO_03
Object-oriented programming. Classes. Class inheritance.	PYPRO_04
Graphic User Interface	PYPRO_05

Functional programming	PYPRO_06

5. Reading list:
   - Hammond, Michael. 2020. Python for Linguists
   - Allen B. Downey: Think Python: How to Think Like a Computer Scientist
     https://www.greenteapress.com/thinkpython/html/index.html

III. Additional information

1. Teaching and learning methods and activities to enable students to achieve the intended course learning outcomes (please indicate the appropriate methods and activities with a tick and/or suggest different methods)

Teaching and learning methods and activities	X
Lecture with a multimedia presentation	X
Interactive lecture	
Problem – based lecture	
Discussions	X
Text-based work	
Case study work	
Problem-based learning	
Educational simulation/game	
Task – solving learning (eg. calculation, artistic, practical tasks)	
Experiential work	X
Laboratory work	
Scientific inquiry method	
Workshop method	
Project work	
Demonstration and observation	
Sound and/or video demonstration	
Creative methods (eg. brainstorming, SWOT analysis, decision tree method, snowball technique, concept maps)	
Group work	X
Other (please specify) -	
…	

2. Assessment methods to test if learning outcomes have been achieved (please indicate with a tick the appropriate methods for each LO and/or suggest different methods)

Assessment methods	Course learning outcome symbol					
	PYPRO_01	PYPRO_02	PYPRO_03	PYPRO_04	PYPRO_05	PYPRO_06
Written exam						
Oral exam						
Open book exam						
Written test						
Oral test						
Multiple choice test	X	X	X	X	X	X
Project	X	X	X	X	X	X
Essay						
Report						
Individual presentation						
Practical exam (performance observation)						
Portfolio						

Other (please specify) -						
…						

3. Student workload and ECTS credits

Activity types		Mean number of hours spent on each activity type
Contact hours with the teacher as specified in the study programme		30
Independent study*	Preparation for classes	30
	Reading for classes	15
	Essay / report / presentation / demonstration preparation, etc.	
	Project preparation	25
	Term paper preparation	
	Exam preparation	
	Other (please specify) -	
	…	
Total hours		100
Total ECTS credits for the course		4

* please indicate the appropriate activity types and/or suggest different activities

4. Assessment criteria in accordance with AMU in Poznan's grading system:

Very good (bdb; 5,0): student knows and understands the concepts very well, can successfully apply them

Good plus (+db; 4,5): student knows and understands the concepts very well, can successfully apply them, but makes small errors

Good (db; 4,0): student knows and understands the concepts well, can successfully apply them, but makes occasional errors

Satisfactory plus (+dst; 3,5): student knows and understands the concepts on a basic level, can apply them on a satisfactory level, but makes errors

Satisfactory (dst; 3,0): student knows and understands the concepts on a basic level, can apply them on a basic level, but makes errors

Unsatisfactory (ndst; 2,0): student does not know or understand the concepts and cannot apply them without making gross errors