

Subject card

Subject name and code	Introduction to Programming, PG_00153241						
Field of study	Mathematical Modeling and Data Analysis						
Date of commencement of studies	October 2025		Academic year of realisation of subject			2025/2026	
Education level	Bachelor's studies		Subject group			Obligatory subject group in the field of study	
Mode of study	full-time studies		Mode of delivery			at the university	
Year of study	1		Language of instruction			Polish	
Semester of study	1		ECTS credits			5.0	
Learning profile	academic		Assessment form			credit	
Conducting unit	Institute of Mathematics -> Faculty of Mathematics, Physics and Informatics -> Rector						
Name and surname of lecturer (lecturers)	Subject supervisor		dr Iwona Krzyżanowska				
	Teachers						
Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	0.0	0.0	60.0	0.0	0.0	60
	E-learning hours included: 0.0						
Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan		Participation in consultation hours		Self-study	SUM
	Number of study hours	60		5.0		60.0	125
Subject objectives	Students acquire the ability to algorithmize simple problems and implement them in a high-level, general-purpose programming language, e.g. Python, based on the basic elements of this language. Developing the ability to control and verify one's own and other people's reasoning and source code. Preparation for in-depth intellectual, literary and technical exploration of IT problems for which simple tools are not sufficient.						

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[MMiADL3_U11] knows how to arrange and analyse an algorithm in accordance with the specification and save it in the selected programming language	The student is able to formulate an algorithm for a simple problem, e.g. mathematics or data analysis, and is able to implement this algorithm in Python by selecting appropriate data types and control instructions. The student is able to design the structure of the program, separating subroutines (functions).	[SU5] implementation of a problem task
	[MMiADL3_W09] knows and understands the basics of computational and programming techniques supporting mathematician's work and understands their limitations	The student knows and understands the basic types of variables, expressions, basic control statements, how to define functions and basic input-output operations in Python. The student understands their limitations.	[SW5] implementation of a problem task
	[MMiADL3_U12] can compile, run and test a self-written computer programme	The student is able to compile, run and test an independently written computer program written in a high-order language in terms of program content.	[SU5] implementation of a problem task [SU8] observation of student's independent or team work
	[MMiADL3_U13] knows how to use computer programmes in the field of data analysis	The student is able to formulate an algorithm for a simple problem, e.g. mathematics or data analysis, and is able to implement this algorithm in Python by selecting appropriate data types and control instructions. The student is able to design the structure of the program, separating subroutines (functions).	[SU5] implementation of a problem task
	[MMiADL3_U09] is able to use the learned software package or the learned programming language to solve selected problems from the known fields, in particular from mathematical analysis, linear algebra and statistics	The student is able to formulate an algorithm for a simple problem, e.g. mathematics or data analysis, and is able to implement this algorithm in Python by selecting appropriate data types and control instructions. The student is able to design the structure of the program, separating subroutines (functions).	[SU5] implementation of a problem task
	[MMiADL3_U10] is able to recognise problems, including practical issues, that can be solved algorithmically; can make a specification of such a problem	The student is able to formulate an algorithm for a simple problem, e.g. mathematics or data analysis, and is able to implement this algorithm in Python by selecting appropriate data types and control instructions. The student is able to design the structure of the program, separating subroutines (functions).	[SU5] implementation of a problem task
Subject contents	[MMiADL3_K03] is ready to work in a team; understands the need for systematic work on all projects that have a long-term character	The student is ready to work in a team and understands the need for systematic involvement in the implementation of projects related to the subject content of a long-term nature.	[SK8] observation of student's independent or team work
	1. Preparing the environment. 2. Basic types of variables (lists, dictionaries, tuples, sets). 3. Creating simple scripts. 4. Conditional if statement. 5. Pseudo-random numbers. 6. For, while loops. 7. Working with a file, writing, reading, modifying. 8. Testing and error handling. 9. Writing your own functions. 10. GUI graphical user interface, e.g. tkinter module, graphics.py 11. Introduction to the classes.		
Prerequisites and co-requisites			
Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	tests and/or project	51.0%	100.0%
	observation of the student's attitude	51.0%	0.0%
Recommended reading	Basic literature	1. https://docs.python.org/3/ 2. Lutz M., Python. Wprowadzenie.	

	Supplementary literature	• Heinold B., A Practical Introduction to Python Programming • McKinney W., Python w analizie danych. Przetwarzanie danych za pomocą pakietów Pandas i NumPy oraz środowiska IPython
	eResources addresses	
Example issues/ example questions/ tasks being completed	none	
Work placement	Not applicable	

Document generated electronically. Does not require a seal or signature.