University of Gdańsk

Subject card

Subject name and code	Programming languages, PG_00143564		
Field of study	Informatics		
Date of commencement of studies	October 2024	Academic year of realisation of subject	2024/2025
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	2	ECTS credits	7.0
Learning profile	academic	Assessment form	exam
Conducting unit			
Name and surname of lecturer (lecturers)	Subject supervisor	dr Tomasz Borzyszkowski	
	Teachers	dr Tomasz Borzyszkowski mgr Laura Grzonka	

Lesson types

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	30.0	0.0	30.0	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	0.0	115.0	175

Subject objectives

The aim of the course is to familiarise students with the principles of structured and object-oriented programming in combination with good programming practices on the basis of the selected programming language.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[INFL3_W05] has general knowledge of various programming paradigms and programming languages; has detailed knowledge of object-oriented design and programming methods and patterns	student explains the operation of basic programming constructs such as selection instructions, loops, blocks, structures, etc.	[SW4] test/exam - oral or written [SW2] presentation/project/paper/ report
[INFL3_U06] is able to design, create, run and test programs using dedicated tools and adequate templates	student is able to apply the programming constructs he/she has learned to write working and correct programs	[SU2] presentation/project/paper/ report [SU8] observation of student's independent or team work
[INFL3_U09] is able to assess the suitability of programming paradigms and tools for solving problems of various types	student is able to apply the programming constructs he/she has learned to write working and correct programs	[SU2] presentation/project/paper/ report [SU8] observation of student's independent or team work

Subject contents	1. Basic concepts:- compilers and interpreters - simple and structured data types and operations on data. 2. Functions: - the definition of a function and its importance in computer program design - methods of passing parameters to a function and returning a result - lambda functions, their mathematical basis and use in a computer program - documentation of functions. 3. Modules and packages: - the concept of namespaces - definition of a module and how to import it - creation of packages/solution libraries. 4. Classes and objects: - class definition and object creation - inheritance and attributes - use of polymorphic functions in object-oriented programming. 5. Unit testing: - concepts of test-driven programming - creation of unit tests- test automation - creating behavioural tests 6. Files and exceptions: - basic file operations serialisation and deserialisation of data - exception creation, raising and handling. 7. Regular expressions:- overview of basic constructs

	- groups and subgroups of regular expressions - using regular expressions for selected programming tasks. 8. Clean code: - an overview of basic programming paradigms and their practical application with selected examples - use of selected design patterns - the concept of technological debt.
Prerequisites and co-requisites	'Introduction to programming' completed

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
exam	50.0%	40.0%
activity in class	0.0%	10.0%
projects	50.0%	50.0%

Recommended reading

Basic literature	Programming methodology: Robert C. Martin, Clean Code. A good programmer's handbook. Helion 2023. Beck Kent, T.D.D. The art of creating good code Helion Publishing House, 2020. Harry J.W. Percival. TDD in practice. Reliable Python code. Helion Publishing House, 2020. Eric Freeman, Elisabeth Freeman, Kathy Sierra, Bert Bates, Head First Design Patterns. Polish edition (Move your head!). Helion 2020.Selected programming language: Guido van Rossum, Python Tutorial, http://docs.python.org/tut/. Mark Pilgrim, Dive into Python. http://diveintopython.org/. Bruce Eckel, Thinking in Python, http://www.mindview.net/Books/TIPython. Python's official documentation, http://docs.python.org/.
Supplementary literature	None
eResources addresses	

Example issues/ example questions/ tasks being completed	
Work placement	Not applicable

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