GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Object-oriented programming, PG_00054485		
Field of study	Automation, Robotics and Control Systems		
Date of commencement of studies	February 2024	Academic year of realisation of subject	2024/2025
Education level	second-cycle studies	Subject group	
Mode of study	Full-time studies	Mode of delivery	at the university
Year of study	2	Language of instruction	Polish
Semester of study	3	ECTS credits	4.0
Learning profile	general academic profile	Assessment form	assessment
Conducting unit	Faculty of Electrical and Control Engineering		
Name and surname of lecturer (lecturers)	Subject supervisor	dr inż. Paweł Kowalski	
	Teachers		

Lesson types and methods of instruction

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	30.0	0.0	15.0	15.0	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	12.0	28.0	100

Subject objectives

Understanding the concepts of object-oriented programming. Acquiring the skills of object modelling, solving problems using objects and relations between them.Acquiring the ability to build and run Java, Python and Kotlin programs with object-oriented programming methods.Acquisition of the ability to design and build a graphical user interface.

Learning outcomes

Course outcome	Subject outcome	Method of verification
K7_U07	Analyzes the operation of a selected website in terms of the data it provides. Designs and builds a web crawler to retrieve data from the website.	[SU1] Assessment of task fulfilment [SU4] Assessment of ability to use methods and tools [SU5] Assessment of ability to present the results of task
K7_W06	Designs and builds a web crawler	[SW2] Assessment of knowledge contained in presentation [SW3] Assessment of knowledge contained in written work and projects
K7_U04	Works independently looking for solutions to the problems encountered in the documentation and on internet forums. Identifies and removes the causes of application malfunctions. Gathering the information necessary to complete the project.	[SU1] Assessment of task fulfilment [SU3] Assessment of ability to use knowledge gained from the subject [SU5] Assessment of ability to present the results of task
K7_W11	Designs and builds applications with a graphical user interface.	[SW2] Assessment of knowledge contained in presentation [SW3] Assessment of knowledge contained in written work and projects
K7_U03	Presents their own solutions to laboratory tasks and developed applications.	[SU1] Assessment of task fulfilment [SU5] Assessment of ability to present the results of task

<table>
<tr><td>Subject contents</td><td>Lecture:

Introduction to object oriented programming and Java. Java data types. Support for standard input and output. Classes and objects. Defining a class and an object. Definitions of fields, methods and constructors. The life cycle of objects and the garbage collector mechanism. Access modifiers visibility of class members. Data Hermetization. Operations on arrays. Inheritance and polymorphism. File handling. Catching, handling and throwing exceptions. Building a GUI application. The essence of event programming. Designing and building mobile applications using the Kotlin language. Data acquisition using an Internet robot.

Laboratory:

During laboratory sessions, the knowledge presented in lectures is practically applied. The laboratories provide an introduction to object-oriented programming techniques in Java, with tasks designed to reinforce key concepts essential for project implementation.

Project:

Development of a system consisting of a graphical user interface application, a web scraper, and a mobile application.</td></tr>
<tr><td>Prerequisites and co-requisites</td><td>Basic programming skills</td></tr>
<tr><td>Assessment methods and criteria</td><td>
<table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>project</td><td>50.0%</td><td>45.0%</td></tr>
<tr><td>exam</td><td>50.0%</td><td>40.0%</td></tr>
<tr><td>laboratory tasks</td><td>50.0%</td><td>15.0%</td></tr>
</table>
</td></tr>
<tr><td>Recommended reading</td><td>
<table>
<tr><td>Basic literature</td><td>• B. Eckel, Thinking in Java, Helion.
• Java Programming Language, Decodejava , https://www.decodejava.com.
• S. Ludwiczak, M. Kunert: Kurs Programowania Java od Podstaw.
• https://javastart.pl/baza-wiedzy,JavaStart, 2021.
• Java Technical Details, http://java.sun.com.</td></tr>
<tr><td>Supplementary literature</td><td>• C. S. Horstmann, G. Cornell: Java. Techniki zaawansowane. Helion, Gliwice 2009.
• A. Redko: Advanced Java Preparing you for Java Mastery, 2015.</td></tr>
<tr><td>eResources addresses</td><td>Adresy na platformie eNauczanie:</td></tr>
</table>
</td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td>• Development of a simple expert system.
• Implementation of the game of life according to the principles of John Conway.
• Designing classes for the passage of a computer game.
• Development of a graphic interface for the selected application.
• Handling of events generated in the designed graphical interface.
• Tic-tac-toe game for mobile devices.
• Development of an Internet robot</td></tr>
<tr><td>Work placement</td><td>Not applicable</td></tr>
</table>

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