University of Gdańsk

Subject card

Subject name and code	Object progamming, PG_00143826		
Field of study	Informatics		
Date of commencement of studies	October 2024	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	part-time studies	Mode of delivery	at the university
Year of study	2	Language of instruction	Polish
Semester of study	3	ECTS credits	8.0
Learning profile	academic	Assessment form	exam
Conducting unit			
Name and surname of lecturer (lecturers)	Subject supervisor	dr Tomasz Borzyszkowski	
	Teachers	dr Tomasz Borzyszkowski mgr Michał Zakrzewski mgr Mateusz Miotk	

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	140.0	200

Subject objectives

The aim of the course is to familiarise students with the principles of object-oriented and event-driven programming on the example of the Java language.To teach students the basic concepts related to object-oriented programming and the ability to construct object-oriented programmes in Java.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[INFL3_U06] is able to design, create, run and test programs using dedicated tools and adequate templates	The student is able to construct object-oriented solutions to simple programming problems using contemporary tools supporting software development.	[SU8] observation of student's independent or team work
[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 knows the basics of object-oriented programming and the basic constructs of the Java language.	[SW4] test/exam - oral or written [SW2] presentation/project/paper/ report
[INFL3_U09] is able to assess the suitability of programming paradigms and tools for solving problems of various types	The student is able to evaluate the suitability of programming paradigms and tools to solve problems of different types	[SU8] observation of student's independent or team work
[INFL3_W10] knows the basic principles of occupational health and safety in the IT profession	The student is familiar with the basic principles of occupational health and safety in the IT profession	[SW1] oral statement/ conversation/discussion

Subject contents	1. Introduction of basic concepts: class, object, field, method; fields and methods: static, public and private in Java; object initialisation and deletion and the so-called rubbish collector mechanism; overview of Java control instructions and their comparison with C control instructions. 2. Java class packages - implementation hiding: package structure; importing packages; setting access rights to package components; construction of interfaces and their implementation. 3. Inheritance and polymorphism: inheritance - syntax and preserving access rights to inherited fields and methods; from abstraction to concrete: abstract and final classes; comparison of properties of final and static fields; comparison of properties of overloaded and polymorphic methods; examples of polymorphic function calls. 4. Overview of classes implementing common data structures: concept of static and dynamic data structures; overview of properties and provided operations on the Java Collection type and derived types (List, Set, BitSet, Maps, Vector, Stack and Hashtable); classes implementing enumeration types and iterators; polymorphic methods allowing sorting of elements stored in collections. 5. Programming with exceptions: overview of predefined exceptions in Java; rules for creating new exceptions; exception reporting and catching. 6. Programming using threads: concept of thread, shared resources and critical section; overview of resource sharing methods in Java: resource locking and the thread jamming problem, and priority queues and the resource access fairness problem; examples of using Java threads to implement classical critical resource access problems. 7. Event-driven programming.
Prerequisites and co-requisites	None
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	1. Eckel B., Thinking in Java - Polish edition. Wydawnictwo HELION, Warszawa, 2010.2 Horstmann C. S., Cornell G., Core Java 2 - Fundamentals. Helion, 2010.3. Horstmann C. S., Cornell G., Core Java 2 - Advanced techniques. Helion, 2010.4. Campione M., Walrath K., Java Tutorial. Addison-Wesley, 2000.
	Supplementary literature	None
	eResources addresses	
Example issues/ example questions/ tasks being completed	1. Dodaj do danej implementacji klasy kalkulatora możliwość przechowywania wyniku w pamięci oraz wykonywania operacji z wykorzystaniem danych przechowywanych w pamięci. 2. Do zadanej implementacji koszyka internetowego dodaj funkcjonalność promocji, korzystając ze wzorce polecenie.	
Work placement	Not applicable	

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