

Subject card

Subject name and code	Object Programming, PG_00191644						
Field of study	Informatics						
Date of commencement of studies	October 2026		Academic year of realisation of subject			2026/2027	
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			5.0	
Learning profile	academic		Assessment form			exam	
Conducting unit							
Name and surname of lecturer (lecturers)	Subject supervisor		dr Tomasz Borzyszkowski				
	Teachers						
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		65.0	125
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
	[INFOL3_W05] knows and understands advanced concepts in the field of various programming paradigms and programming languages; as well as object-oriented design and programming methods and patterns	The student defines and explains advanced mechanisms of object-oriented programming (including encapsulation, interfaces, inheritance and method polymorphism), and describes the structure and purpose of selected design patterns.	[SW2] presentation/project/paper/report
	[INFOL3_U02] is able to use his/her knowledge of higher mathematics to model and solve complex problems	The student is able to model complex algorithmic problems using the object-oriented paradigm, employing logical models to solve classic concurrency problems (e.g. the resource access fairness problem) and selecting appropriate operations on sets and graphs using data structures (e.g. the Set and Map interfaces).	[SU2] presentation/project/paper/report
	[INFOL3_U06] is able to select and apply appropriate IT methods and tools to solve complex problems, is able to use acquired knowledge by appropriately selecting sources and information derived from them, and perform evaluation, critical analysis, and synthesis of this information	The student is able to independently select and implement the most appropriate Java mechanisms – including dynamic data structures (collections), multithreading, and mechanisms for throwing and catching exceptions – in order to implement a secure and stable solution to a complex programming problem.	[SU2] presentation/project/paper/report
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. Overview of selected design patterns.		
Prerequisites and co-requisites			
Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	projects	51.0%	50.0%
	exam	51.0%	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	
Work placement	Not applicable

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