



## Subject card

Subject name and code	Object-oriented programming languages II, PG_00064057						
Field of study	Technical Physics						
Date of commencement of studies	October 2024		Academic year of realisation of subject		2025/2026		
Education level	first-cycle studies		Subject group		Optional subject group Subject group related to scientific research in the field of study		
Mode of study	Full-time studies		Mode of delivery		at the university		
Year of study	2		Language of instruction		Polish		
Semester of study	4		ECTS credits		3.0		
Learning profile	general academic profile		Assessment form		assessment		
Conducting unit	Katedra Fizyki Teoretycznej i Informatyki Kwant. -> Faculty of Applied Physics and Mathematics -> Wydział Politechniki Gdańskiej						
Name and surname of lecturer (lecturers)	Subject supervisor		dr hab. inż. arch. Jan Kozicki				
	Teachers						
Lesson types and methods of instruction	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	0.0	0.0	45.0	0.0	0.0	45
	E-learning hours included: 0.0						
	eNauczanie source addresses: Moodle ID: 35077 Obiektowe języki programowania II 2025/2026 sem.letni <a href="https://enauczanie.pg.edu.pl/moodle/course/view.php?id=35077">https://enauczanie.pg.edu.pl/moodle/course/view.php?id=35077</a>						
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	45		4.0		26.0	75
Subject objectives	Student learns object-oriented programming in the selected programming language (C++ ISO/ANSI, C++14, C++17).						
Learning outcomes	Course outcome		Subject outcome		Method of verification		
	[K6_W05] has knowledge of programming methodology and techniques, and the use of selected IT tools in physics and technology		Has skills in programming in chosen language and can uses basic programming tools.		[SW1] Assessment of factual knowledge		
	[K6_K01] understands the need to learn and improve professional and personal competencies, inspires and organizes other people's learning process		Has basic knowledge in programming and the tools used in physics and technical sciences.		[SK5] Assessment of ability to solve problems that arise in practice		
	[K6_U03] knows programming languages and can use basic software packages		Understands the need to learn whole life and increasing the skills. Can inspire other people.		[SU1] Assessment of task fulfilment		

Subject contents	The basic elements of object-oriented design		
	Reuse of code		
	Analysis of Object		
	Abstract data types		
	Classes and Objects		
	Memory management		
	Mechanisms of inheritance		
	Exception handling		
	Object-oriented design methodology		
	The use of object-oriented techniques in different programming languages		
Prerequisites and co-requisites	Knowledge of operating systems Unix/Linux and MS Windows. Knowledge of the courses Procedural Programming Languages I (FIZ1C301) and II (FIZ1C307). Knowledge of the course Object-Oriented Programming Languages I (FIZ1C305).		
Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	Weekly short assignments based on lecture material from each week.	50.0%	20.0%
	A written knowledge test of the lecture material	50.0%	20.0%
	Test of practical programming skills (C ++ ISO / ANSI).	50.0%	20.0%
	Programming project - C++	50.0%	20.0%
	Very short tests of the practical skills of programming	50.0%	20.0%
Recommended reading	Basic literature	1) B. Stroustrup Programming Principles and Practice using C++, Addison Wesley	
	Supplementary literature	1. B. Meyer Object oriented software construction 2nd Ed.Prientice Hall PTR	
	eResources addresses		

Example issues/ example questions/ tasks being completed	<p>1. Create a vector of Fibonacci numbers and print them using the function from exercise 2. To create the vector, write a function, <code>fibonacci(x,y,v,n)</code>, where integers <code>x</code> and <code>y</code> are ints, <code>v</code> is an empty vector, and <code>n</code> is the number of elements to put into <code>v</code>; <code>v[0]</code> will be <code>x</code> and <code>v[1]</code> will be <code>y</code>. A Fibonacci number is one that is part of a sequence where each element is the sum of the two previous ones. For example, starting with 1 and 2, we get 1, 2, 3, 5, 8, 13, 21, . . . . Your <code>fibonacci()</code> function should make such a sequence starting with its <code>x</code> and <code>y</code> arguments.</p> <p>2. Define an <code>Order</code> class with (customer) name, address, data, and vector members. <code>Purchase</code> is a class with a (product) name, <code>unit_price</code>, and count members. Define a mechanism for reading and writing <code>Orders</code> to and from a file. Define a mechanism for printing <code>Orders</code>. Create a file of at least ten <code>Orders</code>, read it into a vector, sort it by name (of customer), and write it back out to a file. Create another file of at least ten <code>Orders</code> of which about a third are the same as in the first file, read it into a list, sort it by address (of customer), and write it back out to a file. Merge the two files into a third using <code>std::merge()</code>.</p> <p>3. Write a binary search function for a vector (without using the standard one). You can choose any interface you like. Test it. How confident are you that your binary search function is correct? Now write a binary search function for a list. Test it. How much do the two binary search functions resemble each other? How much do you think they would have resembled each other if you had not known about the STL?</p> <p>4. Modify the calculator from Chapter 7 minimally to let it take input from a file and produce output to a file (or use your operating system's facilities for redirecting I/O). Then devise a reasonably comprehensive test for it.</p> <p>5. What are the advantages and disadvantages of intrusive containers compared to C++ standard (non-intrusive) containers? Make lists of pros and cons.</p> <p>6. Make a window (based on <code>My_window</code>) with a 4-by-4 checkerboard of square buttons. When pressed, a button performs a simple action, such as printing its coordinates in an output box, or turns a slightly different color (until another button is pressed).</p> <p>7. explain keywords "this" and "constexpr"</p> <p>8. what is the difference between static polymorphism and dynamic polymorphism. Explain with a code example using keywords "typename" and "virtual".</p>
Work placement	Not applicable

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