GDAŃSK UNIVERSITY
OF TECHNOLOGY

Subject card

Subject name and code	Object-oriented programming languages II, PG_00037343		
Field of study	Technical Physics		
Date of commencement of studies	October 2021	Academic year of realisation of subject	2022/2023
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	blended-learning
Year of study	2	Language of instruction	Polish
Semester of study	4	ECTS credits	5.0
Learning profile	general academic profile	Assessment form	assessment
Conducting unit	Department of Theoretical Physics and Quantum Information -> Faculty of Applied Physics and Mathematics		
Name and surname of lecturer (lecturers)	Subject supervisor	dr hab. inż. arch. Jan Kozicki	
	Teachers	dr hab. inż. arch. Jan Kozicki	

Lesson types and methods of instruction	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	15.0	0.0	45.0	0.0	0.0	60
	E-learning hours included: 57.0						

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	10.0	55.0	125

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_K01	Student can inspire and organize the process of studying.	[SK5] Assessment of ability to solve problems that arise in practice
K6_U03	Student can work with selected IDEs (Integrated Development Environment) for programming in C++	[SU1] Assessment of task fulfilment
K6_W05	Student has advanced programming skills in C++	[SW1] Assessment of factual knowledge

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
Very short tests of the practical skills of programming	50.0%	20.0%
A written knowledge test of the lecture material	50.0%	20.0%
Programming project - C++	50.0%	20.0%
Test of practical programming skills (C ++ ISO / ANSI).	50.0%	20.0%
Weekly short assignments based on lecture material from each week.	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	Uzupełniające Adresy na platformie eNauczanie: Obiektowe języki programowania II 2022/2023 sem.letni - Moodle ID: 27307 https://enauczanie.pg.edu.pl/moodle/course/view.php?id=27307

Example issues/ example questions/ tasks being completed	1. Create a vector of Fibonacci numbers and print them using the functionfrom exercise 2. To create the vector, write a function, fibonacci(x,y,v,n),where integers x and y are ints, v is an empty vector, and n is thenumber of elements to put into v; v[0] will be x and v[1] will be y. A Fibo-nacci number is one that is part of a sequence where each element is thesum of the two previous ones. For example, starting with 1 and 2, we get1, 2, 3, 5, 8, 13, 21, ... Your fibonacci() function should make such asequence starting with its x and y arguments. 2. Define an Order class with (customer) name, address, data, andvector members. Purchase is a class with a (product) name,unit_price, and count members. Define a mechanism for reading andwriting Orders to and from a file. Define a mechanism for printing Or-ders. Create a file of at least ten Orders, read it into a vector, sortit by name (of customer), and write it back out to a file. Create anotherfile of at least ten Orders of which about a third are the same as in the firstfile, read it into a list, sort it by address (of customer), and writeit back out to a file. Merge the two files into a third using std::merge(). 3. Write a binary search function for a vector (without using the standard one). You can choose any interface you like. Test it. How confidentare you that your binary search function is correct? Now write a binarysearch function for a list. Test it. How much do the two binarysearch functions resemble each other? How much do you think theywould have resembled each other if you had not known about the STL? 4. Modify the calculator from Chapter 7 minimally to let it take input froma file and produce output to a file (or use your operating systems facilitiesfor redirecting I/O). Then devise a reasonably comprehensive test for it. 5. What are the advantages and disadvantages of intrusive containers com-pared to C++ standard (non-intrusive) containers? Make lists of prosand cons. 6. Make a window (based on My_window) with a 4-by-4 checkerboard ofsquare buttons. When pressed, a button performs a simple action, such asprinting its coordinates in an output box, or turns a slightly different color(until another button is pressed). 7. explain keywords "this" and "constexpr" 8. what is the difference between static polymorphism and dynamic polymorphism. Explain with a code example using keywords "typename" and "virtual".
Work placement	Not applicable