



Subject card

Subject name and code	Object-oriented programming languages II, PG_00037343						
Field of study	Technical Physics						
Date of commencement of studies	October 2022	Academic year of realisation of subject			2023/2024		
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: 58.0						
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_W05		Has basic knowledge in programming and the tools used in physics and technical sciences.		[SW1] Assessment of factual knowledge		
	K6_K01		Understands the need to learn whole life and increasing the skills. Can inspire other people.		[SK5] Assessment of ability to solve problems that arise in practice		
	K6_U03		Has skills in programming in chosen language and can uses basic programming tools.		[SU1] Assessment of task fulfilment		

Subject contents	<p>The basic elements of object-oriented design</p> <p>Reuse of code</p> <p>Analysis of Object</p> <p>Abstract data types</p> <p>Classes and Objects</p> <p>Memory management</p> <p>Mechanisms of inheritance</p> <p>Exception handling</p> <p>Object-oriented design methodology</p> <p>The use of object-oriented techniques in different programming languages</p>																				
Prerequisites and co-requisites	<p>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).</p>																				
Assessment methods and criteria	<table border="1"> <thead> <tr> <th data-bbox="454 960 794 987">Subject passing criteria</th> <th data-bbox="799 960 1139 987">Passing threshold</th> <th data-bbox="1144 960 1482 987">Percentage of the final grade</th> </tr> </thead> <tbody> <tr> <td data-bbox="454 994 794 1043">Very short tests of the practical skills of programming</td> <td data-bbox="799 994 1139 1043">50.0%</td> <td data-bbox="1144 994 1482 1043">20.0%</td> </tr> <tr> <td data-bbox="454 1050 794 1126">Weekly short assignments based on lecture material from each week.</td> <td data-bbox="799 1050 1139 1126">50.0%</td> <td data-bbox="1144 1050 1482 1126">20.0%</td> </tr> <tr> <td data-bbox="454 1133 794 1182">A written knowledge test of the lecture material</td> <td data-bbox="799 1133 1139 1182">50.0%</td> <td data-bbox="1144 1133 1482 1182">20.0%</td> </tr> <tr> <td data-bbox="454 1189 794 1238">Test of practical programming skills (C ++ ISO / ANSI).</td> <td data-bbox="799 1189 1139 1238">50.0%</td> <td data-bbox="1144 1189 1482 1238">20.0%</td> </tr> <tr> <td data-bbox="454 1245 794 1272">Programming project - C++</td> <td data-bbox="799 1245 1139 1272">50.0%</td> <td data-bbox="1144 1245 1482 1272">20.0%</td> </tr> </tbody> </table>			Subject passing criteria	Passing threshold	Percentage of the final grade	Very short tests of the practical skills of programming	50.0%	20.0%	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%
Subject passing criteria	Passing threshold	Percentage of the final grade																			
Very short tests of the practical skills of programming	50.0%	20.0%																			
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%																			
Recommended reading	<table border="1"> <tbody> <tr> <td data-bbox="454 1292 794 1341">Basic literature</td> <td colspan="2" data-bbox="799 1292 1482 1341">1) B. Stroustrup Programming Principles and Practice using C++, Addison Wesley</td> </tr> <tr> <td data-bbox="454 1348 794 1397">Supplementary literature</td> <td colspan="2" data-bbox="799 1348 1482 1397">1. B. Meyer Object oriented software construction 2nd Ed. Prentice Hall PTR</td> </tr> <tr> <td data-bbox="454 1404 794 1500">eResources addresses</td> <td colspan="2" data-bbox="799 1404 1482 1500"> Adresy na platformie eNauczenie: Obiektowe języki programowania II 2023/2024 sem.letni - Moodle ID: 35076 https://enauczenie.pg.edu.pl/moodle/course/view.php?id=35076 </td> </tr> </tbody> </table>			Basic literature	1) B. Stroustrup Programming Principles and Practice using C++, Addison Wesley		Supplementary literature	1. B. Meyer Object oriented software construction 2nd Ed. Prentice Hall PTR		eResources addresses	Adresy na platformie eNauczenie: Obiektowe języki programowania II 2023/2024 sem.letni - Moodle ID: 35076 https://enauczenie.pg.edu.pl/moodle/course/view.php?id=35076										
Basic literature	1) B. Stroustrup Programming Principles and Practice using C++, Addison Wesley																				
Supplementary literature	1. B. Meyer Object oriented software construction 2nd Ed. Prentice Hall PTR																				
eResources addresses	Adresy na platformie eNauczenie: Obiektowe języki programowania II 2023/2024 sem.letni - Moodle ID: 35076 https://enauczenie.pg.edu.pl/moodle/course/view.php?id=35076																				

<p>Example issues/ example questions/ tasks being completed</p>	<ol style="list-style-type: none"> 1. Create a vector of Fibonacci numbers and print them using the function from 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 the number of elements to put into v; v[0] will be x and v[1] will be y. 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 fibonacci() function should make such a sequence starting with its x and y arguments. 2. Define an Order class with (customer) name, address, data, and vector members. Purchase is a class with a (product) name, unit_price, and count members. Define a mechanism for reading and writing Orders to and from a file. Define a mechanism for printing Orders. Create a file of at least ten Orders, 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 Orders 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 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 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? 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. 5. What are the advantages and disadvantages of intrusive containers compared to C++ standard (non-intrusive) containers? Make lists of pros and cons. 6. Make a window (based on My_window) 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). 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".
<p>Work placement</p>	<p>Not applicable</p>