



Karta przedmiotu

Nazwa i kod przedmiotu	Obiektowe języki programowania II, PG_00037343						
Kierunek studiów	Fizyka Techniczna						
Data rozpoczęcia studiów	październik 2020 r.	Rok akademicki realizacji przedmiotu			2021/2022		
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	2	Język wykładowy			polski		
Semestr studiów	4	Liczba punktów ECTS			5.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Wydział Fizyki Technicznej i Matematyki Stosowanej -> Katedra Fizyki Teoretycznej i Informatyki Kwantowej						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr hab. inż. arch. Jan Kozicki					
	Prowadzący zajęcia z przedmiotu	dr hab. inż. arch. Jan Kozicki					
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	45.0	0.0	0.0	60
	W tym liczba godzin zajęć na odległość: 0.0						
	Adres na platformie eNauczanie: https://enauczanie.pg.edu.pl/moodle/course/view.php?id=9210 Adresy na platformie eNauczanie: Obiektowe języki programowania II 2021/2022 sem.letni - Moodle ID: 19728 https://enauczanie.pg.edu.pl/moodle/course/view.php?id=19728						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów	Udział w konsultacjach		Praca własna studenta		RAZEM
	Liczba godzin pracy studenta	60	10.0		55.0		125
Cel przedmiotu	Nauczenie studenta programowania w wybranych językach obiektowych (C++ ISO/ANSI, C++14, C++17). Przedstawienie korzyści wynikających z posługiwania się technikami zorientowanymi obiektowo.						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu		Sposób weryfikacji i oceny efektu		
	[K6_U03] Posiada umiejętność programowania w wybranym języku oraz stosowania podstawowych pakietów oprogramowania.		Opanuje umiejętność programowania w języku C++ wraz z biblioteką standardową STL.		[SU1] Ocena realizacji zadania		
	[K6_W05] Posiada podstawową wiedzę w zakresie metodyki i technik programowania oraz wykorzystywania wybranych narzędzi informatycznych w fizyce i technice.		Opanuję podstawową wiedzę niezbędną (stosowane metody oraz techniki programowania, np. komentowanie kodu) do programowania i wykorzystywania wybranych narzędzi informatycznych w fizyce i technice.		[SW1] Ocena wiedzy faktograficznej		
	[K6_K01] Rozumie potrzebę uczenia się przez całe życie oraz potrzebę podnoszenia kompetencji zawodowych i osobistych. Potrafi inspirować i organizować proces uczenia się innych osób.		Student rozumie potrzebę szukania nowych i lepszych bibliotek programistycznych. Używania nowych technologii, śledzenie rozwoju języka C++17 i przyszłego standardu C++21.		[SK5] Ocena umiejętności rozwiązywania problemów występujących w praktyce		

Treści przedmiotu	<ul style="list-style-type: none"> • Podstawowe elementy projektowania obiektowego • Wielokrotne wykorzystywanie kodu • Analiza obiektowa • Abstrakcyjne typy danych • Klasy i obiekty • Zarządzanie pamięcią • Mechanizmy dziedziczenia • Obsługa wyjątków • Metodyka projektowania obiektowego • Zastosowanie technik obiektowych w różnych językach programowania 																				
Wymagania wstępne i dodatkowe	Znajomość systemów operacyjnych Unix/Linux i MS Windows. Wiedza z przedmiotów Proceduralne Języki Programowania I (FIZ1C301) i II (FIZ1C307) oraz Obiektowe Języki Programowania I (FIZ1C305).																				
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	<table border="1" data-bbox="451 387 1487 712"> <thead> <tr> <th data-bbox="451 387 794 421">Sposób oceniania (składowe)</th> <th data-bbox="794 387 1145 421">Próg zaliczeniowy</th> <th data-bbox="1145 387 1487 421">Składowa oceny końcowej</th> </tr> </thead> <tbody> <tr> <td data-bbox="451 421 794 477">Pisemny sprawdzian wiedzy z wykładu.</td> <td data-bbox="794 421 1145 477">50.0%</td> <td data-bbox="1145 421 1487 477">20.0%</td> </tr> <tr> <td data-bbox="451 477 794 510">Projekt programistyczny - C++</td> <td data-bbox="794 477 1145 510">50.0%</td> <td data-bbox="1145 477 1487 510">20.0%</td> </tr> <tr> <td data-bbox="451 510 794 589">Sprawdzian z praktycznej umiejętności programowania (C++ ISO/ANSI).</td> <td data-bbox="794 510 1145 589">50.0%</td> <td data-bbox="1145 510 1487 589">20.0%</td> </tr> <tr> <td data-bbox="451 589 794 645">Cotygodniowo zadawane krótkie zadania z bieżącego materiału.</td> <td data-bbox="794 589 1145 645">50.0%</td> <td data-bbox="1145 589 1487 645">20.0%</td> </tr> <tr> <td data-bbox="451 645 794 712">Krótkie sprawdziany ("wejściówki") na początku zajęć laboratoryjnych.</td> <td data-bbox="794 645 1145 712">50.0%</td> <td data-bbox="1145 645 1487 712">20.0%</td> </tr> </tbody> </table>			Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej	Pisemny sprawdzian wiedzy z wykładu.	50.0%	20.0%	Projekt programistyczny - C++	50.0%	20.0%	Sprawdzian z praktycznej umiejętności programowania (C++ ISO/ANSI).	50.0%	20.0%	Cotygodniowo zadawane krótkie zadania z bieżącego materiału.	50.0%	20.0%	Krótkie sprawdziany ("wejściówki") na początku zajęć laboratoryjnych.	50.0%	20.0%
Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej																			
Pisemny sprawdzian wiedzy z wykładu.	50.0%	20.0%																			
Projekt programistyczny - C++	50.0%	20.0%																			
Sprawdzian z praktycznej umiejętności programowania (C++ ISO/ANSI).	50.0%	20.0%																			
Cotygodniowo zadawane krótkie zadania z bieżącego materiału.	50.0%	20.0%																			
Krótkie sprawdziany ("wejściówki") na początku zajęć laboratoryjnych.	50.0%	20.0%																			
Zalecana lista lektur	<table border="1" data-bbox="451 723 1487 902"> <tbody> <tr> <td data-bbox="451 723 794 768">Podstawowa lista lektur</td> <td colspan="2" data-bbox="794 723 1487 768">1) B. Stroustrup Programming Principles and Practice using C++, Addison Wesley</td> </tr> <tr> <td data-bbox="451 768 794 801">Uzupełniająca lista lektur</td> <td colspan="2" data-bbox="794 768 1487 801">1. B. Meyer Programowanie zorientowane obiektowo, Helion</td> </tr> <tr> <td data-bbox="451 801 794 902">Adresy eZasobów</td> <td colspan="2" data-bbox="794 801 1487 902">Obiektowe języki programowania II 2021/2022 sem.letni - Moodle ID: 19728 https://enauczanie.pg.edu.pl/moodle/course/view.php?id=19728</td> </tr> </tbody> </table>			Podstawowa lista lektur	1) B. Stroustrup Programming Principles and Practice using C++, Addison Wesley		Uzupełniająca lista lektur	1. B. Meyer Programowanie zorientowane obiektowo, Helion		Adresy eZasobów	Obiektowe języki programowania II 2021/2022 sem.letni - Moodle ID: 19728 https://enauczanie.pg.edu.pl/moodle/course/view.php?id=19728										
Podstawowa lista lektur	1) B. Stroustrup Programming Principles and Practice using C++, Addison Wesley																				
Uzupełniająca lista lektur	1. B. Meyer Programowanie zorientowane obiektowo, Helion																				
Adresy eZasobów	Obiektowe języki programowania II 2021/2022 sem.letni - Moodle ID: 19728 https://enauczanie.pg.edu.pl/moodle/course/view.php?id=19728																				
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	<p>1. Stwórz wektor liczb Fibonacciego i wypisz je stosując funkcję z ćwiczenia 2 (str.300). Aby stworzyć wektor napisz funkcję fibonacci(x,y,v,n), w której dwie początkowe liczby x i y są typu int, v to pusty std::vector, and n to liczba elementów. Każdy kolejny element w sekwencji to suma dwóch poprzednich, przykładowo, startując z 1 i 2 otrzymamy: 1, 2, 3, 5, 8, 13, 21, . . . Twoja funkcja powinna tworzyć taką sekwencję z dwóch początkowych liczb x,y.</p> <p>2. Zdefiniuj klasę Zlecenie zawierającą imię (zleceniodawcy), adres, datę i std::vector. Zakup to jest klasa zawierająca nazwę produktu, cenę oraz ilość. Zdefiniuj mechanizm czytania i pisania do pliku do klasy Zlecenie. Zdefiniuj funkcję drukującą zlecenia na ekranie. Stwórz plik zawierający co najmniej 10 zleceń, wczytaj go do std::vector, posortuj go według imienia zleceniodawcy i zapisz z powrotem do pliku. Zdefiniuj drugi plik zawierający co najmniej 10 zleceń z których około jedna trzecia to takie same zlecenia jak w pierwszym pliku, wczytaj go do std::list, posortuj go po adresie zleceniodawcy i zapisz z powrotem do pliku. Następnie połącz oba pliki używając polecenia std::merge()</p> <p>3. Napisz funkcję wyszukiwania binarnego dla std::vector (bez używania funkcji std::find() z biblioteki standardowej). Możesz stworzyć do tego dowolny interfejs jaki uznasz za stosowny. Przetestuj ją. Czy jesteś pewien, że Twoja funkcja działa prawidłowo? Teraz napisz funkcję wyszukiwania binarnego dla std::list. Przetestuj ją. Na ile te funkcje są podobne? Zastanów się czy można je zastąpić jedną funkcją z użyciem szablonów (templates). Czy dałoby się to zrobić gdybyś nie wiedział o istnieniu STL?</p> <p>4. Zmodyfikuj kalkulator z rozdziału 7 tak aby mógł on pobierać dane wejściowe z pliku i zapisywać wynik do pliku (lub skorzystaj z przekierowywania strumieni w systemie operacyjnym). Następnie skonstruuj w miarę dokładny test tego kalkulatora.</p> <p>5. Jakie są wady i zalety intruzywnych kontenerów w porównaniu ze standardowymi nie-intruzywnymi kontenerami C++? Zrób listę was i zalet.</p> <p>6. Zrób okienko (oparte na przykładzie My_window ze str.578) zawierające szachownicę 4x4 z kwadratowymi przyciskami. Przycisk po przyciśnięciu powinien wykonywać jakąś prostą akcję, np. wypisać współrzędne w okienku output lub zmienia swój kolor do czasu aż nie zostanie wciśnięty kolejny przycisk.</p> <p>7. wyjaśnić słowa kluczowe "this" i "constexpr"</p> <p>8. wyjaśnić na przykładzie kodu czym się różni polimorfizm statyczny od polimorfizmu dynamicznego, posłużyć się przy tym słowami kluczowymi "typename" oraz "virtual".</p>																				
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy																				