

KARTA PRZEDMIOTU (SYLABUS)

Opis przedmiotu

Kod przedmiotu		Nazwa przedmiotu	PROGRAMOWANIE OBIEKTOWE	
I/O/1/NST/B1-5			OBJECT-ORIENTED PROGRAMMING	
Język wykładowy		język polski		
Rok akademicki		2024/2025		
Kierunek		Informatyka		
w zakresie		-		
Poziom studiów		studia pierwszego stopnia		
Profil studiów		ogólnoakademicki		
Forma studiów		studia niestacjonarne		
Semestr / semestry		2		
Przynależność do grupy zajęć		B1. Grupa zajęć kierunkowych – obowiązkowych		
Status przedmiotu		obowiązkowy		
Formy realizacji zajęć dydaktycznych, wymiar, punkty ECTS		Forma zajęć	Liczba godzin zajęć dydaktycznych	Liczba punktów ECTS
		Wykład	18 [h]	5 ECTS
		Laboratorium	18 [h]	
Powiązanie przedmiotu	z profilem studiów	związany z prowadzoną działalnością naukową w dyscyplinach, do których przyporządkowany jest kierunek studiów		3 ECTS
	z uprawnieniami	służy do zdobywania przez studenta kompetencji inżynierskich		4 ECTS
	z dyscypliną	Informatyka Techniczna i Telekomunikacja		5 ECTS
Forma nauczania		tradycyjna – zajęcia zorganizowane w Uczelni i/lub zajęcia z wykorzystaniem metod i technik kształcenia na odległość (max. 0,72 ECTS)		
Wymagania wstępne				
Jednostka prowadząca		Katedra Informatyki i Teleinformatyki		
Koordynator		dr Artur Bartoszewski		
Adres strony internetowej pjo		www.wteii.uniwersytetradom.pl		
Adres e-mail, telefon koordynatora		artur.bartoszewski@urad.edu.pl, +48 48 3617806		

EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE, REALIZACJA ZAJĘĆ DYDAKTYCZNYCH, WERYFIKACJA EFEKTÓW UCZENIA SIĘ

Cel kształcenia:		Zapoznanie studentów z fundamentalnymi koncepcjami programowania obiektowego w języku C++, a także z praktycznymi aspektami jego zastosowania. Studenci zdobędą umiejętności niezbędne do tworzenia kompleksowych programów obiektowych w C++.			
Treści programowe:		<p>Wykład [BN, W1, W2, U1, U2, K1]:</p> <ol style="list-style-type: none"> 1. Podstawowe pojęcia programowania obiektowego, klasy i składniki klas; Dostęp; Enkapsulacja; Klasa, a obiekt. 2. Konstruktor i destruktor klasy. Wskaźnik this, Przeciążanie metod. 3. Przekazywanie obiektów do funkcji. Metody przekazywania obiektów, różnice między przekazywaniem przez wartość a referencję. 4. Składniki statyczne klas. 5. Dziedziczenie, typy dziedziczenia, mechanizmy dziedziczenia w C++. 6. Funkcje zaprzyjaźnione. 7. Konstruktor kopiujący; Koncepcja płytkiego i głębokiego kopiowania obiektów. 8. Przeciążanie operatorów. 9. Polimorfizm i klasy wirtualne. 10. Standardowa biblioteka C++ (STL) - kontenery: vector, list, map. 11. Szablony (Templates) - szablony funkcji i szablony klas, ich definicja i zastosowanie. 12. Praktyczne programowanie obiektowe - przykłady kompleksowych programów obiektowych demonstrujących zastosowanie omawianych koncepcji. <p style="text-align: right;">Suma: 18 [h]</p> <p>Laboratorium [BN, U1, U2, K1]:</p> <ol style="list-style-type: none"> 1. Wprowadzenie do programowania obiektowego, definicja klasy w C++, składniki klas; Dostęp; Enkapsulacja. 2. Konstruktor i destruktor. 3. Dynamiczne i statyczne tworzenie obiektów, tablice obiektów i tablice wskaźników do obiektów. 4. Wskaźnik this, Przeciążanie metod. 5. Przekazywanie obiektów do funkcji. 			

	6. Składniki statyczne klas, Funkcje zaprzyjaźnione; 7. Przykłady praktyczne (np. klasa realizująca kolejkę LIFO, klasa która zapisująca do pliku tekstowego informacje o czasie utworzenia i usunięcia obiektów itp.) 8. Wykorzystanie mechanizmu dziedziczenia. Strategie dziedziczenia. 9. Konstruktor kopiujący; Płytkie i głębokie kopiowanie obiektów. 10. Przeciążanie operatorów, Przykład praktyczny (np. klasa umożliwiająca wykonywanie działań na wektorach). 11. Polimorfizm i klasy wirtualne. 12. Zastosowanie standardowej biblioteka C++ (STL) - kontenery: vector, list, map. Suma: 18 [h]
Metody dydaktyczne (kształcenia):	– metody podające (wykład informacyjny) – metody problemowe (wykład problemowy, wykład konwersatoryjny) – metody aktywizujące (metoda przypadków, dyskusja dydaktyczna), – metody praktyczne (pokaz, ćwiczenia laboratoryjne).
Rygor zaliczenia, kryteria oceny osiągniętych efektów uczenia się, sposób obliczania oceny końcowej:	Warunkiem zaliczenia przedmiotu jest osiągnięcie wszystkich wymaganych efektów uczenia się określonych dla danego przedmiotu. Uzyskanie pozytywnych ocen ze wszystkich form zajęć wchodzących w skład danego przedmiotu jest równoznaczne z jego zaliczeniem i zdobyciem przez studenta liczby punktów ECTS przyporządkowanej temu przedmiotowi. Sposób obliczenia oceny końcowej z przedmiotu określa regulamin studiów. Sposób obliczania oceny z poszczególnych form zajęć przedstawia się następująco: Na ocenę z laboratorium składa się ocena z kolokwium praktycznego. Na ocenę z wykładu składa się wynik egzaminu pisemnego. Ocena z egzaminu - zdobyte punkty przeliczane zostają na ocenę wg skali: Ocena 2 poniżej 51% Ocena 3 od 51% Ocena 3,5 od 61% Ocena 4 od 71% Ocena 4,5 od 81% Ocena 5 od 91%

Efekty uczenia się dla przedmiotu w odniesieniu do efektów kierunkowych i formy zajęć				Metody weryfikacji efektów uczenia się	
Numer efektu uczenia się	Opis efektów uczenia się dla przedmiotu (PEU) Student, który zaliczył przedmiot (W) zna i rozumie/ (U) potrafi /(K) jest gotów do:	Kierunkowy efekt uczenia się (KEU)	Forma zajęć	Forma weryfikacji (zaliczeń)	Metody sprawdzania i oceny
W1	obiektyowy paradygmat programowania	K_WG06 K_WG08	wykład, ćwiczenia laboratoryjne	egzamin, zaliczenie na ocenę	egzamin, kolokwium
W2	metody programowania obiektowego w języku C++	K_WG06	wykład	egzamin, zaliczenie na ocenę	egzamin, kolokwium
U1	projektować i implementować programy komputerowe z wykorzystaniem paradygmatu obiektowego	K_UW04	wykład	egzamin, zaliczenie na ocenę	egzamin, kolokwium
U2	dobierać właściwe struktury danych oraz konstrukcje algorytmiczne do rozwiązywanego problemu oraz umie zaprogramować algorytm w formie obiektowej	K_UW04 K_UW05	wykład, ćwiczenia laboratoryjne	egzamin, zaliczenie na ocenę	egzamin, kolokwium
K1	zaplanować pracę pod kątem zakładanych rezultatów, określić priorytetowe zadania w oparciu o zasady skutecznego działania.	K_KO05	ćwiczenia laboratoryjne	zaliczenie na ocenę	kolokwium

Literatura i pomoce naukowe
1. Grębosz J.: Symfonia C++, Programowanie w języku C++ orientowane obiektowo, Wydawnictwo Edition 2000. 2. Grębosz J. : Opus magnum C++. Programowanie w języku C++. Wydanie III poprawione, Helion 2024. 3. Stroustrup B.: Język C++. Kompendium wiedzy. Wydanie IV, Helion 2014 4. Eckel B.: Thinking in C++. Edycja polska, Helion, Gliwice 2002. 5. Walczak K.: Nauka programowania obiektowego w języku C++, W & W Warszawa 2002. 6. Chomicz P.: Programowanie w języku C i C++. PLI 1992. 7. Grębosz J.: Pasja C++, Wydawnictwo Edition 2000. 8. Meyers S.: Język C++ bardziej efektywnie, Wydawnictwo Naukowo Techniczne, Warszawa 1996.

Nakład pracy studenta potrzebny do osiągnięcia zakładanych efektów uczenia się – bilans punktów ECTS			
Udział w zajęciach, aktywność	Obciążenie studenta [h]		
	Inne godz. kontaktowe (IGK)	Zajęcia bez nauczyciela-praca własna studenta (ZBN)	Zajęcia dydaktyczne
Udział w wykładach	X	X	18 [h]
Udział w ćwiczeniach / laboratoriach / projektach / seminariach	X	X	18 [h]
Udział w konsultacjach	6 [h]	X	X
Przygotowanie do wykładów / ćwiczeń / laboratoriów / projektów / seminariów	X	83 [h]	X
Przygotowanie do zaliczenia/egzaminu			
Sumaryczne obciążenie pracą studenta	6 [h] / 0,2 ECTS	83 [h] / 3,4 ECTS	36 [h] / 1,4 ECTS
Punkty ECTS za przedmiot	5 ECTS		

Informacje dodatkowe, uwagi
<p>W przypadku studentów ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekle chorych, określone powyżej (w karcie) metody i formy weryfikacji efektów uczenia się dostosowuje się odpowiednio do indywidualnych potrzeb tych studentów.</p> <p>Szczegółowe zasady i formy wsparcia studentów ze szczególnymi potrzebami: w tym z niepełnosprawnością, przewlekle chorych podczas zajęć, zaliczeń i egzaminów określono w: Regulaminie Studiów, Zasadach Studiowania, Procedurze dotyczącej zapewnienia dostępności procesu kształcenia studentom ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekle chorych.</p>