

Karta przedmiotu

Nazwa i kod przedmiotu	Podstawy programowania - algorytmy i struktury danych, PG_00062716						
Kierunek studiów	Technologie Przemysłu 5.0						
Data rozpoczęcia studiów	październik 2025 r.		Rok akademicki realizacji przedmiotu		2025/2026		
Poziom kształcenia	I stopnia - inżynierskie		Grupa zajęć		Grupa zajęć obowiązkowych z zakresu kierunku studiów		
Forma studiów	stacjonarne		Sposób realizacji		na uczelni		
Rok studiów	1		Język wykładowy		polski		
Semestr studiów	1		Liczba punktów ECTS		6.0		
Profil kształcenia	ogólnoakademicki		Forma zaliczenia		zaliczenie		
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Inżynierii Biomedycznej						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr inż. Grzegorz Jasiński				
	Prowadzący zajęcia z przedmiotu		dr inż. Grzegorz Jasiński				
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	0.0	30.0	0.0	0.0	60
	W tym liczba godzin zajęć na odległość: 0.0						
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		5.0		85.0	150
Cel przedmiotu	Celem przedmiotu jest nabycie przez studentów wiedzy oraz umiejętności w zakresie programowania. Student powinien opanować umiejętność tworzenia i analizy algorytmów oraz zasady tworzenia oprogramowania w języku C/C++: instrukcji, typów danych, operatorów i funkcji. Student powinien nabyć wiedzę na temat struktur, wskaźników oraz innych podstawowych pojęć związanych z programowaniem w języku C/C++.						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu		Sposób weryfikacji i oceny efektu		
	[K6_K01] jest świadoma potrzeby stałego aktualizowania i wzbogacania posiadanej wiedzy i umiejętności praktycznych, podnoszenia kompetencji zawodowych, osobistych i społecznych		Student rozumie potrzebę poznawania i uczenia się dobrze znanych technologii. Student potrafi analizować problemy i dobrać właściwe modele danych oraz struktury danych. Student rozumie cykl życia i wytwarzania kodu .		[SK2] Ocena postępów pracy [SK5] Ocena umiejętności rozwiązywania problemów występujących w praktyce [SK3] Ocena umiejętności organizacji pracy		
	[K6_W01] wykazuje się znajomością i zrozumieniem matematyki, fizyki, chemii oraz narzędzi informatycznych na poziomie niezbędnym do formułowania i rozwiązywania typowych problemów inżynierskich oraz technologicznych		Student zna podstawowe struktury danych i odpowiadające im algorytmy, Student potrafi pisać programy w języku C/C++ z wykorzystaniem odpowiednich bibliotek.		[SW1] Ocena wiedzy faktograficznej		
	[K6_U01] stosuje wiedzę z matematyki, fizyki, chemii, narzędzi informatycznych i innych dyscyplin inżynierskich do rozwiązywania problemów teoretycznych, inżynierskich oraz technologicznych		Student programuje w języku programowania proceduralnego, uruchamia i testuje programy. Student potrafi analizować problemy i dobrać właściwe modele danych oraz struktury danych.		[SU2] Ocena umiejętności analizy informacji [SU4] Ocena umiejętności korzystania z metod i narzędzi		

Treści przedmiotu	1. Języki programowania, alfabet, syntaktyka i semantyka. Translacja. 2. Klasyfikacja typów. Typy całkowite i zmiennoprzecinkowe. 3. Operatory i wyrażenia arytmetyczne. 4. Wybrane funkcje standardowe. 5. Typ znakowy. Rzutowanie typów. 6. Typ logiczny. Operatory i wyrażenia logiczne. 7. Podstawy obsługi wejścia/wyjścia. 8. Instrukcje warunkowe (if, switch) oraz wyrażenie warunkowe. 9. Instrukcje iteracyjne (for, while, do-while). Iteracje zagnieżdżone. 10. Definiowanie typów. Stałe. Typ wyliczeniowy. 11. Tablice jednowymiarowe i wielowymiarowe. Napisy. 12. Zakres ważności i czas życia zmiennych 13. Funkcje. 14. Przesyłanie parametrów funkcji. 15. Typ wskaźnikowy. Arytmetyka wskaźników. 16. Wskaźniki w komunikacji między funkcjami. 17. Dynamiczna alokacja pamięci. 18. Struktury 19. Pojęcie algorytmu, różnice między algorytmami, wpływ modelu na rozwiązanie, reprezentacja algorytmów, analiza poprawności. 20. Przykłady rozwiązań rekurencyjnych i iteracyjnych, metody wyczerpujące, heurystyczne, programowanie dynamiczne. 21. Przykłady rekurencyjnych rozwiązań typu dziel i rządź. 22. Podstawowe struktury danych: lista, kolejka, stos i metody ich implementacji. 23. Podstawowe algorytmy sortowania, poszukiwanie binarne, sortowanie szybkie, kopcowe.											
Wymagania wstępne i dodatkowe												
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	<table><tr><td>Sposób oceniania (składowe)</td><td>Próg zaliczeniowy</td><td>Składowa oceny końcowej</td></tr><tr><td>Wykład</td><td>50.0%</td><td>50.0%</td></tr><tr><td>Laboratorium</td><td>50.0%</td><td>50.0%</td></tr></table>	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej	Wykład	50.0%	50.0%	Laboratorium	50.0%	50.0%		
Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej										
Wykład	50.0%	50.0%										
Laboratorium	50.0%	50.0%										
Zalecana lista lektur	Podstawowa lista lektur	KERNIGHAN, Brian W.; RITCHIE, Dennis M. The C programming language, Prentice Hall, 2006 Grębosz Jerzy, Symfonia C++ Standard (tom 1 i 2), Wydanie 2000, Krakow 2008										
	Uzupełniająca lista lektur	Stephen Prata, "Język C++. Szkoła programowania". Wydanie VI. Helion 2012 Mirosław J. Kubiak, "C++. Zadania z programowania z przykładowymi rozwiązaniami", Helion 2011										
	Adresy eZasobów	Adresy na platformie eNauczanie:										
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	Napisanie programu realizującego zadaną funkcjonalność. Przeanalizowanie sposobu działania zadanego programu.											
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy											

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.