POLITECHNIKA GDAŃSKA

Karta przedmiotuuuu

Nazwa i kod przedmiotu	Programming, PG_00064129		
Kierunek studiów	Elektronika i telekomunikacja (studia w jęz. angielskim), Informatyka (studia w jęz. angielskim), Automatyka, cybernetyka i robotyka (studia w jęz. angielskim)		
Data rozpoczęcia studiów	październik 2025 r.	Rok akademicki realizacji przedmiotu	2025/2026
Poziom kształcenia	II stopnia	Grupa zajęć	Grupa zajęć obowiązkowych z zakresu kierunku studiów
Forma studiów	stacjonarne	Sposób realizacji	na uczelni
Rok studiów	1	Język wykładowy	angielski
Semestr studiów	1	Liczba punktów ECTS	7.0
Profil kształcenia	ogólnoakademicki	Forma zaliczenia	zaliczenie
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Algorytmów i Modelowania Systemów		
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr Marcin Jurkiewicz	
	Prowadzący zajęcia z przedmiotu	dr Marcin Jurkiewicz	

Formy zajęć

Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
Liczba godzin zajęć	30.0	0.0	30.0	20.0	0.0	80

W tym liczba godzin zajęć na odległość: 0.0

Adresy kursu na platformie eNauczanie:

Moodle ID: 46792 2025 Programming ABC
https://enauczanie.pg.edu.pl/moodle/course/view.php?id=46792

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	80	11.0	84.0	175

Cel przedmiotu

Celem przedmiotu jest nabycie przez studentów wiedzy oraz umiejętności w zakresie programowania i implementacji programów w środowisku Linux/Visual Studio. Student powinien opanować podstawowy repertuar dostępnych w języku C/C++ instrukcji, typów i struktur danych, operatorów, funkcji oraz związanych z nimi algorytmów. 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
[K7_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i technik programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów, dokonując oceny i krytycznej analizy wykonanego oprogramowania, a także syntezy i twórczej interpretacji prezentowanych za jego pomocą informacji	Student potrafi wykorzystać przekazaną wiedzę (z wykładu), podstawowe techniki języka C/C++ oraz oprogramowanie w systemie Linux/Visual Studio do napisania i kompilacji programu.	[SU1] Assessment of task fulfilment [SU4] Assessment of ability to use methods and tools
[K7_W04] zna i rozumie w pogłębionym stopniu zasady, metody i techniki programowania oraz zasady tworzenia oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo inne elementy lub układy programowalne, specyficznych dla kierunku studiów, a także organizację pracy systemów wykorzystujących komputery lub te urządzenia	Student zna podstawowe zasady programowania języka C/C++ oraz struktury danych i algorytmy.	[SW1] Assessment of factual knowledge

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. Zasięg i czas życia zmiennych. Efekt uboczny.
14. Przesyłanie parametrów funkcji.
15. Typ wskaźnikowy. Arytmetyka wskaźników.
16. Wskaźniki w komunikacji między funkcjami.
17. Dynamiczna alokacja pamięci.
18. Struktury
19. Podstawy złożoności obliczeniowej i notacja O
20. Problemy wielomianowe i NP trudne
21. Algorytmy sortowania
22. Kopce
23. Kolejka i stos
24. Haszowanie
25. Algorytmy zachłanne, heurystyki i pełnego przeglądu

Wymagania wstępne i dodatkowe

Brak wymagań

Sposoby i kryteria oceniania osiąganych efektów uczenia się

Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
Projekt: poprawność działania programu, algorytmy, struktury danych, czas wykonania oraz uniwersalność programu.	50.0%	30.0%
Laboratorium: poprawność działania programu, algorytmy, struktury danych, czas wykonania oraz uniwersalność programu.	50.0%	30.0%
Egzamin	50.0%	40.0%

Zalecana lista lektur

Podstawowa lista lektur	1. KERNIGHAN, Brian W.; RITCHIE, Dennis M. *The C programming language*, Prentice Hall, 2006
Uzupełniająca lista lektur	1. B. Stroustrup, *The C++ Programming Language*, Addison Wesley Longman, 2000
Adresy eZasobów	

Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	**Co pokaże poniższy fragment kodu, jeśli będzie częścią poprawnego programu?** int i; for(i=0;i<3;i++); cout << i; cout << i+1; a) 011223 b) 0124 c) 0123 d) 34 e) 124 f) 45
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.