POLITECHNIKA GDAŃSKA

Karta przedmiotu

Nazwa i kod przedmiotu	INFORMATYKA I, PG_00038090		
Kierunek studiów	Automatyka, robotyka i systemy sterowania		
Data rozpoczęcia studiów	październik 2024 r.	Rok akademicki realizacji przedmiotu	2024/2025
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć	
Forma studiów	stacjonarne	Sposób realizacji	na uczelni
Rok studiów	1	Język wykładowy	polski
Semestr studiów	2	Liczba punktów ECTS	4.0
Profil kształcenia	ogólnoakademicki	Forma zaliczenia	zaliczenie
Jednostka prowadząca	Wydział Elektrotechniki i Automatyki		
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr inż. Robert Smyk	
	Prowadzący zajęcia z przedmiotu	dr inż. Daniel Wachowiak dr inż. Paweł Kowalski dr inż. Robert Smyk	

Formy zajęć i metody nauczania

Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
Liczba godzin zajęć	30.0	0.0	15.0	0.0	0.0	45

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	45	10.0	45.0	100

Cel przedmiotu

Celem ogólnym przedmiotu jest wprowadzenie studentów w podstawowe zagadnienia programowania oraz zapoznanie ich z kluczowymi technikami przetwarzania danych i tworzenia aplikacji komputerowych. Kurs ma na celu rozwinięcie umiejętności programistycznych poprzez naukę podstaw języka **C** oraz **Python**, a także przedstawienie podstawowych zasad programowania obiektowego i wykorzystywania interfejsów graficznych (GUI).

Cele szczegółowe

- Opanowanie podstawowej składni i struktur języka **C** oraz **Python**.
- Nauka stosowania **tablic**, **funkcji** i **dynamicznej alokacji pamięci** w języku C.
- Zrozumienie zasad **programowania obiektowego** i wykorzystania **klas oraz obiektów** w Pythonie.
- Umiejętność **przetwarzania danych ustrukturyzowanych** i zarządzania plikami.
- Implementacja podstawowych algorytmów i struktur danych.
- Tworzenie aplikacji z **interfejsem graficznym (GUI)**.
- Poznanie metod organizacji kodu i modularności programów.
- Rozwój umiejętności analitycznych i algorytmicznych.
- Wykorzystanie narzędzi do debugowania i optymalizacji kodu.
- Przygotowanie do dalszej nauki zaawansowanych technik programowania.

Efekty uczenia się przedmiotu

Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
[K6_U04] ma umiejętność samokształcenia się m.in. w celu podnoszenia kwalifikacji zawodowych	Potrafi rozwiązań zadanie programistyczne z wykorzystaniem pętli, warunków.	[SU4] Ocena umiejętności korzystania z metod i narzędzi
[K6_W06] zna strukturę komputerów i mikroprocesorów oraz zadania systemów operacyjnych, ma podstawową wiedzę z podstaw oprogramowania komputerów, sterowników, techniki mikroprocesorowej, projektowania prostych algorytmów oraz działania sieci informatycznych	Potrafi zaprogramować wybrany algorytm sortowania lub wyszukiwania.	[SW1] Ocena wiedzy faktograficznej

Treści przedmiotu	**Podstawy programowania w języku C** • Składnia i struktura programów w C. • Typy danych, operatory i instrukcje sterujące. • Tablice jednowymiarowe i wielowymiarowe. • Funkcje i ich zastosowanie. • Dynamiczna alokacja pamięci i zarządzanie zasobami. **Podstawy programowania w języku Python** • Składnia i struktura programów w Pythonie. • Typy danych, operatory i instrukcje sterujące. • Listy, krotki, słowniki i zbiory. • Obsługa plików i operacje wejścia/wyjścia. **Programowanie obiektowe w Pythonie** • Klasy, obiekty, metody i atrybuty. • Dziedziczenie • Moduły i pakiety w Pythonie. **Przetwarzanie danych ustrukturyzowanych** • Operacje na danych tekstowych i binarnych. • Obsługa plików CSV, JSON i XML. • Podstawowe operacje na bazach danych (SQLite). **Tworzenie interfejsów graficznych (GUI)** • Wprowadzenie do bibliotek GUI (Tkinter, PyQt, inne). • Projektowanie i implementacja okien aplikacji. • Obsługa zdarzeń użytkownika. **Algorytmy i struktury danych** • Podstawowe algorytmy sortowania i wyszukiwania. • Listy i tablice. • Efektywność algorytmów. **Zarządzanie kodem i debugowanie** • Podstawy pracy z systemem kontroli wersji. • Debugowanie kodu i obsługa błędów. • Optymalizacja kodu i dobre praktyki programistyczne.
Wymagania wstępne i dodatkowe	
Sposoby i kryteria oceniania osiąganych efektów uczenia się	

Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
Sprawdziany praktyczne	60.0%	37.5%
Wejściówki	60.0%	12.5%
Sprawdziany teoretyczne	60.0%	37.5%
Prace domowe	60.0%	12.5%

Zalecana lista lektur	Podstawowa lista lektur	1. B. Kernighan, D. Ritchie, Język C, WNT 1988. 2. Niklaus Wirth, Algorytmy + struktury danych = programy, WNT 1989. 3. William Stallings, Computer Organization And Architecture. Designing for performance. 8th-edition.
	Uzupełniająca lista lektur	brak
	Adresy eZasobów	Adresy na platformie eNauczanie:
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	Zamieniać do innych postaci liczby w postaciach dziesiętnych, binarnych, szesnastkowych i ósemkowych Wymienić zadania systemu operacyjnego Wyjaśnić różnice między rekurencyjnym a iteracyjnym sposobem programowania Opisać zasady analizy złożoności algorytmów Przedstawić działanie wybranych algorytmów sortowania Przedstawić podejścia do tworzenia oprogramowanie w wielkiej skali, i różnice między nimi Tworzenie programów w języku C realizujących określone zadania i wykorzystujące przedstawione techniki programowania: - programy obliczeniowe - prosta gra komputerowa - przetwarzanie łańcuchów znaków	
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy	