

<b>WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI</b>	
<b>KARTA PRZEDMIOTU</b>	
<b>Nazwa przedmiotu w języku polskim: Podstawy programowania</b>	
<b>Nazwa przedmiotu w języku angielskim: Programming principles</b>	
<b>Kierunek studiów (jeśli dotyczy): Informatyka Techniczna</b>	
<b>Specjalność (jeśli dotyczy): .....</b>	
<b>Poziom studiów:</b>	<b>I stopień</b>
<b>Forma studiów:</b>	<b>stacjonarna</b>
<b>Rodzaj przedmiotu:</b>	<b>obowiązkowy</b>
<b>Język wykładowy:</b>	<b>polski</b>
<b>Cykl kształcenia od:</b>	<b>2024/2025</b>
<b>Kod przedmiotu</b>	<b>W04ITE-SI0037G</b>
<b>Grupa kursów</b>	<b>TAK</b>

	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
Liczba godzin zajęć zorganizowanych w Uczelni (ZZU)	30	15	15		
Liczba godzin całkowitego nakładu pracy studenta (CNPS)	40	30	30		
Forma zaliczenia (egzamin lub zaliczenie na ocenę)	zaliczenie na ocenę	zaliczenie na ocenę	zaliczenie na ocenę		
Dla grupy kursów zaznaczyć kurs końcowy (X)	X				
Liczba punktów ECTS	4				
w tym liczba punktów odpowiadająca zajęciom o charakterze praktycznym (P)	-	1	2		
w tym liczba punktów ECTS odpowiadająca zajęciom wymagającym bezpośredniego udziału nauczycieli lub innych osób prowadzących zajęcia (BU)	1,36	0,68	0,68		

**WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I KOMPETENCJI SPOŁECZNYCH**

**CELE PRZEDMIOTU**

- C1 Nabycie podstawowej wiedzy z zakresu algorytmów komputerowych oraz sposobów ich przedstawiania i analizowania.
- C2 Poznanie podstawowych konstrukcji programistycznych wspólnych dla większości języków algorytmicznych: typów, zmiennych, warunkowych rozgałęzień, pętli, funkcji z argumentami, rekurencji, tablic, list, plików.
- C3 Nabycie umiejętności programowania strukturalnego i proceduralnego w języku C lub C++.
- C4 Poznanie standardowych algorytmów przetwarzania dużych ilości danych: przeszukiwania, agregowania i sortowania.
- C5 Zapoznanie się z wybranymi formami dynamicznych i złożonych struktur danych: listą, stosem,

kolejką, drzewem.

C6 Nabycie umiejętności konfigurowania i posługiwania się wybranymi środowiskami programistycznymi w celu usprawnienia procesów edycji, kompilacji i testowania projektów programistycznych.

### PRZEDMIOTOWE EFEKTY UCZENIA SIĘ

#### Z zakresu wiedzy:

- PEU\_W01 Posiada podstawową wiedzę na temat nowoczesnych języków i paradygmatów programowania.
- PEU\_W02 Zna język reprezentacji oraz zasady konstruowania schematów blokowych
- PEU\_W03 Zna składnię i typowe konstrukcje programistyczne języka C lub C++.
- PEU\_W04 Zna zasady programowania strukturalnego i proceduralnego.
- PEU\_W05 Rozumie pojęcia: iteracji, rekurencji, organizacji pamięci, arytmetyki wskaźników oraz dynamicznego rezerwowania i zwalniania zasobów.
- PEU\_W06 Zna podstawowe algorytmy wyszukiwania, agregowania i sortowania danych.
- PEU\_W07 Posiada wiedzę na temat wybranych dynamicznych i złożonych struktur danych.
- PEU\_W08 Zna narzędzia programistyczne wspomagające pracę informatyka.

#### Z zakresu umiejętności:

- PEU\_U01 Umie zapisać algorytm w postaci schematu blokowego.
- PEU\_U02 Potrafi skonstruować rozwiązanie prostych zadań programistycznych wymagających użycia kilku rozgałęzień, pętli lub rekurencji.
- PEU\_U03 Umie zdefiniować funkcję oraz dobrać sposób przekazywania parametrów wejściowych i wyniku działania funkcji.
- PEU\_U04 Potrafi definiować, inicjalizować oraz przetwarzać podstawowe reprezentacje danych: tablice, łańcuchy znakowe, struktury oraz ich kombinacje.
- PEU\_U05 Umie poprawnie strukturalizować kod oraz dane programu w języku C lub C++, zgodnie z zasadami programowania strukturalnego i proceduralnego.
- PEU\_U06 Potrafi oprogramować operacje przechowywania danych w pamięci trwałej wykorzystując strumienie plikowe.
- PEU\_U07 Potrafi wykorzystywać wskaźniki i instrukcje alokacji do dynamicznego zarządzania pamięcią wykorzystywaną przez program.
- PEU\_U08 Potrafi zaprojektować i oprogramować zestaw funkcji ukrywających szczegóły implementacyjne wybranych złożonych i dynamicznych struktur danych.
- PEU\_U09 Potrafi zaproponować oraz przeprowadzić procedurę symbolicznego lub dynamicznego testowania poprawności wykonanego oprogramowania.
- PEU\_U10 Umie wykorzystać zintegrowane środowisko programistyczne do skonfigurowania, edytowania i testowania projektów jednowątkowych programów konsolowych.
- PEU\_U11 Potrafi pozyskiwać informacje dotyczące programowania z dokumentacji technicznej, literatury, Internetu oraz innych źródeł w języku polskim i angielskim.

#### Z zakresu kompetencji społecznych:

- PEU\_K01 Rozumie potrzebę ustawicznego poszerzania swojej wiedzy, w tym systematycznego zapoznawania się z nowymi publikacjami z zakresu informatyki i dokumentacją nowych produktów.
- PEU\_K02 Jest świadom prawnych i społecznych aspektów informatyzacji oraz potrzeby przestrzegania zasad etycznych w działalności zawodowej informatyka.

### TREŚCI PROGRAMOWE

Forma zajęć - wykład		Liczba godzin
Wy1	Algorytmy i sposoby ich przedstawiania. Dominujące paradygmaty programowania. Język schematów blokowych. Etapy i narzędzia wykorzystywane podczas tworzenia oprogramowania. Standardy języków programowania. Ogólna struktura programu w języku C lub C++. Przykłady	2

	kodów źródłowych programów konsolowych oraz podstawowe konstrukcje programowe.	
Wy2	Dane i ich komputerowe reprezentacje. Typy danych i zakresy ich wartości. Zmienne programowe, deklaracje zmiennych i inicjowanie wartości. Zasięg widoczności identyfikatorów. Klasy pamięci. Identyfikatory typów (typedef). Operatory i wyrażenia: arytmetyczne, relacyjne, logiczne, bitowe. Obliczanie wartości wyrażeń algebraicznych. Standardowe funkcje matematyczne. Podstawowe operacje wejścia/wyjścia oraz dialog z użytkownikiem w trybie znakowym. Formatowane wejście i wyjście z wykorzystaniem standardowych bibliotek <stdio.h> <iostream>.	2
Wy3	Podstawowe instrukcje: przypisania, warunkowa i wyboru. Sterowanie wykonaniem programu, składanie i zagnieżdżanie instrukcji rozgałęziających. Przykłady algorytmów przetwarzających nieduże ilości danych (bez wykorzystania pętli). Pojęcie iteracji w programie. Rodzaje pętli: while, do while, for. Warunki zakończenia pętli i zagnieżdżanie pętli. Instrukcje break i continue. Proste algorytmy iteracyjne: zliczanie, sumowanie i poszukiwanie ekstremum w ciągu danych pobieranych ze strumienia.	2
Wy4	Tablice w języku C/C++: deklaracja oraz inicjalizacja, dostęp do elementów za pomocą operatora indeksu. Operacje na tablicach z wykorzystaniem pętli for. Tablice wielowymiarowe. Podstawowe algorytmy przetwarzania tablic.	2
Wy5	Funkcje i procedury w językach programowania. Pojęcia: prototypu, definicji i wywołania funkcji. Funkcje bezparametrowe. Zwracanie wartości funkcji. Jawne przekazywanie danych przez listę argumentów. Przekazywanie argumentów przez wartość i przez referencję. Argumenty domniemane. Funkcje przeciążone. Funkcje inline. Funkcje rekurencyjne.	2
Wy6	Wskaźniki zmiennych i ich adresy, arytmetyka wskaźników. Związek pomiędzy wskaźnikami a tablicami. Praca z tablicami w zapisie wskaźnikowym. Przekazywanie argumentów funkcji przez adres. Funkcje standardowe operujące bezpośrednio na pamięci (memset, memcpy, memcmp, memmove, itp.)	2
Wy7	Tablicowa reprezentacja tekstów w języku C/C++. Standardowe funkcje łańcuchowe z biblioteki <string.h> (strcpy, strcmp, strcat, strlen, itd.). Przykłady pisania własnych funkcji przetwarzających dane tekstowe.	2
Wy8	Kolokwium połówkowe (formujące) Specyfikacja programu, testowanie, obsługa błędów, dokumentowanie.	2
Wy9	Rekurencja i algorytmy rekurencyjne. Przeszukiwanie binarne i sortowanie tablic.	2
Wy10	Typ strukturalny - pojęcie struktury w języku C/C++. Definicja, deklaracja i inicjalizacja zmiennych strukturalnych. Zagnieżdżanie typów złożonych (struktur i tablic). Przykład prostej bazy danych wykorzystującej reprezentację w postaci tablic struktur.	2
Wy11	Obsługa plików zewnętrznych. Pliki o dostępie swobodnym i pliki tekstowe. Proceduralne i obiektowe biblioteki operacji plikowych. Standardowe funkcje do obsługi plików z biblioteki <stdio.h>. Wejście i wyjście dla znaków, łańcuchów i danych formatowanych. Wejście i wyjście blokowe (binarne). Przenaszalność danych pomiędzy różnymi systemami operacyjnymi.	2
Wy12	Dynamiczne przydzielanie pamięci. Alokacja i zwalnianie pamięci przydzielonej dynamicznie (funkcje malloc, calloc, free, operatory new i delete). Kontrola zajętości serty. Dynamiczne tworzenie i realokacja tablic oraz łańcuchów znaków o zadawanej wielkości	2
Wy13	Złożone struktury wskaźnikowe. Tablica wskaźników na zmienne proste,	2

	tablica wskaźników na tablice / łańcuchy o stałej wielkości, dynamiczna tablica wskaźników na dynamiczne łańcuchy. Wskaźniki na funkcje. Funkcja qsort.	
Wy14	Tworzenie dynamicznych struktur danych: lista wskaźnikowa, stos, kolejka, kolejka priorytetowa, drzewa binarne i ich własności.	2
Wy15	Repetytorium. Kolokwium zaliczeniowe.	2
	Suma godzin	<b>30</b>

<b>Forma zajęć - ćwiczenia</b>		<b>Liczba godzin</b>
Ćw1	Omówienie programu oraz organizacji zajęć ćwiczeniowych. Zapis algorytmów za pomocą języka schematów blokowych.	1
Ćw2	Reprezentacja danych różnego typu. Dobór typu zmiennych, ograniczenia reprezentacji. Dialog z użytkownikiem z wykorzystaniem printf i scanf Formatowanie danych (budowa łańcuchów formatujących zawierających różnorodne sekwencje sterujące % \ ) Zapis wyrażeń matematycznych w języku C/C++. Zapis wyrażeń logicznych (operatory logiczne)	2
Ćw3	Pojęcie iteracji. Rola i dobór zmiennych sterujących oraz pomocniczych pętli. Budowanie warunków końca pętli. Algorytmy iteracyjne (zliczanie, sumowanie, maksimum, minimum, obliczanie szeregów). Równoważność pętli. Programowanie proceduralne - podział zadania na podprogramy-funkcje, menu sterujące. Zakres widoczności i przesłanianie identyfikatorów.	2
Ćw4	Podstawowe algorytmy przetwarzania tablic (wypełnianie, porównywanie elementów, wyszukiwanie, przesuwanie, usuwanie, dodawanie elementów) Tablica pseudo-dynamiczna (statyczna tablica z licznikiem wykorzystywanych elementów). Parametryzacja algorytmów. Dobór sposobu przekazywania argumentów wejściowych oraz wyników funkcji.	2
Ćw5	Funkcje przetwarzające teksty. Analiza funkcji z biblioteki <string.h>. Oprogramowanie własnych funkcji przetwarzających łańcuchy znaków w. Dynamiczna alokacja i realokacja pamięci – tablice jednowymiarowe o zmiennym rozmiarze. Arytmetyka wskaźników, konwersja (rzutowanie) wskaźników. Ćwiczenia z dostępu do dowolnego obszaru pamięci.	2
Ćw6	Strukturalna dekompozycja dużych programów oraz złożonych reprezentacji danych. Omówienie i ćwiczenia z reprezentacją problemu prostej bazy danych za pomocą tablicy struktur. Kodowanie danych "nienumerycznych" - typ wyliczeniowy. Kodowanie danych za pomocą słownika. Operacje składowania danych w pamięci zewnętrznej za pomocą strumieni plikowych. Tekstowa i binarna reprezentacja danych liczbowych. Wykrywanie błędów operacji wej/wyj. Sterowanie położeniem wskaźnika pliku. Podstawowe algorytmy sekwencyjnego przetwarzania plików tekstowych i binarnych.	2
Ćw7	Analiza wzorcowych implementacji złożonych-dynamicznych struktur danych: listy wskaźnikowej, stosu, kolejki, kolejki priorytetowej. Analiza wzorcowych implementacji wybranych rekurencyjnych algorytmów sortowania tablic.	2
Ćw8	Repetytorium. Zaliczenia.	2
	Suma godzin	<b>15</b>

<b>Forma zajęć - laboratorium</b>		<b>Liczba godzin</b>
La1	Omówienie programu oraz organizacji zajęć laboratoryjnych. Szkolenie stanowiskowe BHP. Konfiguracja środowiska programistycznego (np. Windows/Visual Studio lub Linux/Emacs/gcc). Przykład programu konsolowego z użyciem zmiennych prostych, instrukcji przypisania i konsolowe operacje wejścia wyjścia. Edycja, kompilacja, uruchomienie i	1

	debugowanie programu.	
La2	Ćwiczenia z tworzeniem programów ilustrujących zastosowanie podstawowych instrukcji i konstrukcji programowych języka C/C++: przypisania, rozgałęzienia warunkowego (if , if/else), wyboru (switch, case, break, default). Zagnieżdżanie instrukcji rozgałęziających. Obliczanie wyrażeń matematycznych.	2
La3	Ćwiczenia z tworzeniem programów ilustrujących zastosowanie instrukcji pętlowych (while, do while, for). Standardowe algorytmy iteracyjne: zliczanie, sumowanie, szukanie maksimum i minimum. Ćwiczenia z tworzeniem własnych funkcji. Funkcje bezparametrowe i zmienne lokalne. Przekazywanie parametrów przez zmienne globalne.	2
La4	Ćwiczenia z tworzeniem programów ilustrujących wykorzystanie reprezentacji tablicowej. Przetwarzanie tablic za pomocą pętli. Wybrane algorytmy przetwarzania tablic: wyszukiwanie liniowe i binarne, sortowanie bąbelkowe i przez wstawianie. Funkcje z jawną listą argumentów. Przekazywanie argumentów przez wartość, referencję i adres. Debugowanie i testowanie poprawności programów.	2
La5	Ćwiczenia z tworzeniem programów ilustrujących przetwarzanie danych tekstowych reprezentowanych w postaci tablicy znaków. Dostęp do zmiennych za pomocą wskaźników. Programy wykorzystujące dynamiczną alokację i realokację tablic jednowymiarowych. Debugowanie i testowanie poprawności programów.	2
La6	Oprogramowanie prostej bazy danych wykorzystującej reprezentację w postaci tablicy struktur lub tablicy wskaźników na struktury. Rozbudowanie programu o operacje archiwizacji danych w pamięci zewnętrznej w postaci plików tekstowych lub binarnych.	2
La7	Oprogramowanie wybranej dynamicznej struktury danych: listy wskaźnikowej, kolejki, kolejki priorytetowej lub drzewa. Ćwiczenia z tworzeniem programów wykorzystujących rekurencję.	2
La8	Repetytorium. Zaliczenia.	2
	Suma godzin	<b>15</b>

### STOSOWANE NARZĘDZIA DYDAKTYCZNE

- N1. Wykład tradycyjny z wykorzystaniem wideoprojektora lub telekonferencji (np. Zoom).  
N2. Praca własna – przygotowanie do ćwiczeń poprzez rozwiązywanie zadań  
N3. Praca własna – samodzielne wykonanie zadanych programów laboratoryjnych  
N4. Inspekcje kodu wykonanych programów przez prowadzącego laboratorium  
N5. Praca własna – samodzielne studia i przygotowanie do kolokwium  
N6. Konsultacje

### OCENA OSIĄGNIĘCIA PRZEDMIOTOWYCH EFEKTÓW UCZENIA SIĘ

Oceny (F – formująca (w trakcie semestru), P – podsumowująca (na koniec semestru))	Numer efektu uczenia się	Sposób oceny osiągnięcia efektu uczenia się
F1	PEU_W01 – W08	(Wykład) Pisemne kolokwium na wykładzie. W przypadku przeprowadzenia dodatkowego kolokwium w połowie semestru, ocena F1 jest sumą ważoną ( $1/3 \cdot F4 + 2/3 \cdot F5$ ) ocen: F4 – z pierwszego kolokwium, F5 – z drugiego kolokwium
F2	PEU_U01 – U02, PEU_U08 – U09,	(Ćwiczenia) Ocena odpowiedzi ustnych. Ocena rozwiązań przykładowych zadań ćwiczeniowych.

	PEU_U11, PEU_K01 – K02	Kolokwium zaliczeniowe na ćwiczeniach.
F3	PEU_U03 – U07, PEU_U10	Obserwacja wykonywania ćwiczeń laboratoryjnych. Inspekcja i ocena kodu wykonanych programów, przez prowadzącego laboratorium.
P = 1/2*F1 + 1/4*F2 + 1/4*F3; oceny składowe muszą być pozytywne: F1≥3.0, F2≥3.0, F3≥3.		

## **LITERATURA PODSTAWOWA I UZUPEŁNIAJĄCA**

### **LITERATURA PODSTAWOWA:**

- [1] Brian W Kernighan, Język ANSI C: programowanie, Helion,
- [2] Kochan S.G., Język C : kompendium wiedzy, Helion,
- [3] Deitel P.J., Język C : solidna wiedza w praktyce, Helion,
- [4] Grębosz J., Symfonia C++ Standard, Editions 2000
- [5] Stroustrup B., Język C++, WNT,
- [6] Wróblewski P., Algorytmy, struktury danych i techniki programowania. Helion

### **LITERATURA UZUPEŁNIAJĄCA:**

- [1] Wirth N., Algorytmy + struktury danych = programy, WNT,
- [2] Segewick C., Algorytmy w C++. W.N.-T., Warszawa
- [3] Prata S., Język C : szkoła programowania, Helion,
- [4] Lippman S. B., Lajoie J., Podstawy języka C++, WNT, Warszawa
- [5] Eckel B., Thinking in C++, Helion,
- [6] Neapolitan R., Podstawy algorytmów z przykładami w C++. Wyd. Helion,

**NAUCZYCIEL AKADEMICKI ODPOWIEDZIALNY ZA PRZEDMIOT (imię,  
nazwisko, adres e-mail)**

**Dr inż. Marek Piasecki, [marek.piasecki@pwr.edu.pl](mailto:marek.piasecki@pwr.edu.pl)**