

WYDZIAŁ ELEKTRONIKI

KARTA PRZEDMIOTU

Nazwa przedmiotu w języku polskim: **Wstęp do programowania**
 Nazwa przedmiotu w języku angielskim: **Introduction to Programming**
 Kierunek studiów (jeśli dotyczy): **Electronic and Computer Engineering**
 Specjalność (jeśli dotyczy):
 Poziom i forma studiów: **I stopień/ stacjonarna**
 Rodzaj przedmiotu: **obowiązkowy**
 Kod przedmiotu: **ECEA00002**
 Grupa kursów: **TAK**

	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
Liczba godzin zajęć zorganizowanych w Uczelni (ZZU)	30		45		
Liczba godzin całkowitego nakładu pracy studenta (CNPS)	120		120		
Forma zaliczenia	zaliczenie na ocenę		zaliczenie na ocenę		
Dla grupy kursów zaznaczyć kurs końcowy (X)	X				
Liczba punktów ECTS	8				
w tym liczba punktów odpowiadająca zajęciom o charakterze praktycznym (P)			4		
w tym liczba punktów ECTS odpowiadająca zajęciom wymagającym bezpośredniego kontaktu (BU)	2		4		

*niepotrzebne skreślić

WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI
CELE PRZEDMIOTU

- C1 Nabycie podstawowej wiedzy z zakresu algorytmów komputerowych oraz sposobów ich przedstawiania i analizowania. Poznanie standardowych algorytmów przetwarzania dużych ilości danych: przeszukiwania, agregowania i sortowania.
- C2 Poznanie podstawowych konstrukcji programistycznych wspólnych dla większości języków algorytmicznych: typów, zmiennych, warunkowych rozgałęzień, pętli, funkcji z argumentami, rekurencji, tablic, list, plików. Zapoznanie się z wybranymi formami dynamicznych i złożonych struktur danych: listą, stosem, kolejką, drzewem.
- C3 Nabycie umiejętności programowania strukturalnego i proceduralnego w języku C lub C++, oraz posługiwania się wybranymi środowiskami programistycznymi w celu usprawnienia procesów edycji, kompilacji i testowania wieloplikowych projektów programistycznych.
- C4 Uaktualnienie i ujednolicenie wiedzy z zakresu technologii informacyjnych oraz doskonalenie umiejętności w zakresie zaawansowanego ich wykorzystywania w pracy zawodowej inżyniera.

PRZEDMIOTOWE EFEKTY UCZENIA SIĘ

Z zakresu wiedzy:

- PEU_W01 Posiada podstawową wiedzę na temat nowoczesnych języków i paradygmatów programowania. Zna język reprezentacji oraz zasady konstruowania schematów blokowych. Zna podstawowe algorytmy wyszukiwania, agregowania i sortowania danych.
- PEU_W02 Zna składnię i typowe konstrukcje programowania strukturalnego i proceduralnego w języku C lub C++. Rozumie pojęcia: iteracji, rekurencji, organizacji pamięci, arytmetyki wskaźników oraz dynamicznego rezerwowania i zwalniania zasobów. Posiada wiedzę na temat wybranych dynamicznych i złożonych struktur danych.
- PEU_W03 Zna narzędzia programistyczne, technologie informatyczne oraz pakiety oprogramowania użytkowego wspomagające pracę informatyka.

Z zakresu umiejętności:

- PEU_U01 Umie zapisać algorytm w postaci schematu blokowego. Potrafi skonstruować rozwiązanie prostych zadań programistycznych wymagających użycia kilku rozgałęzień, pętli lub rekurencji.
- PEU_U02 Umie poprawnie strukturalizować kod oraz dane programu w języku C/C++, zgodnie z zasadami programowania strukturalnego i proceduralnego. Umie zdefiniować funkcję oraz dobrać sposób przekazywania parametrów wejściowych i wyniku działania funkcji. Potrafi definiować, inicjalizować oraz przetwarzać podstawowe reprezentacje danych: tablice, łańcuchy znakowe, struktury oraz ich kombinacje.
- PEU_U03 Potrafi wykorzystywać wskaźniki i instrukcje alokacji do dynamicznego zarządzania pamięcią wykorzystywaną przez program. Potrafi zaprojektować i oprogramować zestaw funkcji ukrywających szczegóły implementacyjne wybranych złożonych i dynamicznych struktur danych. Potrafi oprogramować operacje przechowywania danych w pamięci trwalej wykorzystując strumienie plikowe.
- PEU_U04 Umie wykorzystać zintegrowane środowisko programistyczne do skonfigurowania, edytowania i testowania projektów jednowątkowych programów konsolowych.
- PEU_U05 Posiada umiejętności w zakresie efektywnego wykorzystywania oprogramowania typu „Office” do tworzenia dokumentacji o złożonej strukturze, arkuszy kalkulacyjnych automatyzujących wykonywanie obliczeń inżynierskich oraz oprogramowania własnych funkcji lub makr.

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 kodów źródłowych programów konsolowych oraz podstawowe konstrukcje programowe.	2
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 wyrażen 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: biblioteka <mem.h> (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 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). Standardy eksportowania/importowania plików tekstowo-numerycznych umożliwiające wymianę danych pomiędzy programami w języku C/C++ a popularnymi pakietami biurowymi.	2
Wy12	Dynamiczne przydzielanie pamięci. Alokacja i zwalnianie pamięci przydzielonej dynamicznie (funkcje malloc, calloc, free, operatory new i delete). Kontrola zajętości sterty. Dynamiczne tworzenie i realokacja tablic oraz łańcuchów znaków o zadawanej wielkości.	2
Wy13	Złożone struktury wskaźnikowe. Tablica wskaźników na zmienne proste, tablica wskaźników na tablice o stałej wielkości, dynamiczna tablica wskaźników na dynamiczne łańcuchy. Wskaźniki na funkcje. Funkcja qsort.	2
Wy14	Zintegrowane pakiety biurowe. Możliwości zaawansowanego przetwarzania dokumentów tekstowych i danych w postaci arkuszy kalkulacyjnych, poprzez samodzielne programowanie nowych funkcji i makr.	2
Wy15	Repetitorium oraz kolokwium zaliczeniowe	2
	Suma godzin	30

Forma zajęć – laboratorium		Liczba godzin
La1	Omówienie programu oraz organizacji zajęć laboratoryjnych. Szkolenie stanowiskowe BHP. Zapis algorytmów za pomocą języka schematów blokowych. Konfiguracja wybranego środowiska programistycznego (np. Windows/Visual Studio lub Linux/Emacs/gcc). Przykład programu konsolowego z użyciem zmiennych prostych, instrukcji przypisania oraz konsolowe operacje wejścia wyjścia. Edycja, kompilacja, uruchomienie i debugowanie programu. Wykorzystanie zintegrowanych pakietów biurowych do sporządzania dokumentacji technicznej oraz sprawozdań z wykonania zadań laboratoryjnych.	3
La2	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)	3
La3	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.	3
La4	Pojęcie iteracji. Rola i dobór zmiennych sterujących oraz pomocniczych pętli. Budowanie warunków końca pętli. Algorytmy iteracyjne (zliczanie, sumowanie, maksimum, minimum, obliczanie szeregów). Równoważność pętli.	3
La5	Ciąg dalszy ćwiczeń z tworzeniem programów ilustrujących zastosowanie instrukcji pętlowych (while, do while, for). Standardowe algorytmy iteracyjne: zliczania, sumowania, szukania maksimum i minimum oraz obliczania szeregów matematycznych.	3
La6	Programowanie proceduralne. Podział zadania na podprogramy-funkcje oraz menu sterujące. Zakres widoczności i przesłanianie identyfikatorów. Ćwiczenia z tworzeniem własnych funkcji. Funkcje bezparametrowe i zmienne lokalne. Przekazywanie parametrów przez zmienne globalne. Funkcje z jawną listą argumentów. Przekazywanie argumentów przez wartość, referencję i adres.	3
La7	Ćwiczenia z tworzeniem programów ilustrujących wykorzystanie reprezentacji tablicowej. Przetwarzanie tablic za pomocą pętli. Podstawowe algorytmy przetwarzania tablic (wypełnianie, porównywanie elementów, wyszukiwanie, przesuwanie, usuwanie, dodawanie elementów)	3
La8	Tablice dynamiczne (z licznikiem wykorzystywanych elementów). Wybrane algorytmy przetwarzania tablic: wyszukiwanie liniowe i binarne, sortowanie bąbelkowe i przez wstawianie. Parametryzacja algorytmów. Dobór sposobu przekazywania argumentów wejściowych oraz wyników funkcji.	3
La9	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.	3
La10	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.	3
La11	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.	3
La12	Strukturalna dekompozycja dużych programów oraz złożonych reprezentacji	3

	danych. Omówienie i ćwiczenia z reprezentacją problemu prostej bazy danych za pomocą tablicy struktur. Kodowanie danych "nienumerycznych" - typ wyliczeniowy. Kodowanie danych za pomocą słownika.	
La13	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. Eksport danych numeryczno-tekstowych do programu arkusza kalkulacyjnego.	3
La14	Oprogramowanie wybranej dynamicznej struktury danych: listy wskaźnikowej, kolejki, kolejki priorytetowej lub drzewa. Ćwiczenia z tworzeniem programów wykorzystujących rekurencję.	3
La15	Ćwiczenia z technikami zaawansowanego formatowania dokumentów technicznych oraz obliczeń inżynierskich z wykorzystaniem arkuszy kalkulacyjnych. Automatyzacja pracy poprzez programowanie własnych funkcji oraz makr.	3
	Suma godzin	45

STOSOWANE NARZĘDZIA DYDAKTYCZNE

- N1. Wykład tradycyjny z wykorzystaniem wideoprojektora.
N2. Praca własna – samodzielne wykonanie zadanych programów laboratoryjnych
N3. InsPEUcje kodu wykonanych programów przez prowadzącego laboratorium
N4. Praca własna – samodzielne studia i przygotowanie do kolokwium z wykładu

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 – PEU_W03	Pisemne kolokwium na wykładzie. W przypadku przeprowadzenia dodatkowego kolokwium w połowie semestru, ocena F1 jest sumą ważoną $(1/3 \cdot F3 + 2/3 \cdot F4)$ ocen: F3 – z pierwszego kolokwium, F4 – z drugiego kolokwium
F2	PEU_U01 – PEU_U05	Ocena sprawozdań dokumentujących wykonane ćwiczenia laboratoryjne. InsPEUcja kodu wykonanych programów z udziałem prowadzącego laboratorium.
$P = 0.4 \cdot F1 + 0.6 \cdot F2$, wszystkie oceny składowe muszą być pozytywne		

LITERATURA PODSTAWOWA I UZUPEŁNIAJĄCA
<u>LITERATURA PODSTAWOWA:</u> [1] Brian Kernighan, Dennis Ritchie, The C Programming Language, 1988 [2] Greg Perry, Dean Miller, C Programming Absolute Beginner's Guide, 3rd Edition, 2013 [3] Bjarne Stroustrup, The C++ programming language, 4th ed., 2013 [4] Stanley Lippman, Josée Lajoie, C++ primer, 5th ed., 2013 <u>LITERATURA UZUPEŁNIAJĄCA:</u> [1] Niklaus Wirth, Algorithms + Data Structures = Programs, 1976 [2] Robert Sedgewick, Algorithms in C, 3rd Edition, 2001 [3] K.N. King, C Programming: A Modern Approach, 1996 [4] Dan Gookin, C for Dummies, Volume 1, 1994 [5] Alex Allain, Jumping into C++2013
<u>OPIEKUN PRZEDMIOTU (IMIE, NAZWISKO, ADRES E-MAIL)</u>
Dr inż. Marek Piasecki, marek.piasecki@pwr.wroc.pl