

POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2020/2021

Wydział Informatyki i Telekomunikacji

Kierunek studiów: Informatyka

Profil: Ogólnoakademicki

Forma studiów: niestacjonarne

Kod kierunku: I

Stopień studiów: I

Specjalności: Brak specjalności

1 INFORMACJE O PRZEDMIOCIE

NAZWA PRZEDMIOTU	Języki i paradygmaty programowania I
NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM	Języki i paradygmaty programowania I
KOD PRZEDMIOTU	WiIT I oIN C11 20/21
KATEGORIA PRZEDMIOTU	Przedmioty kierunkowe
LICZBA PUNKTÓW ECTS	4.00
SEMESTRY	2

2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁAD	ĆWICZENIA	LABORATORIUM	LABORATORIUM KOMPUTERO- WE	SEMINARIUM	PROJEKT
2	18	0	18	0	0	0

3 CELE PRZEDMIOTU

Cel 1 Poznanie podstawowych paradygmatów programowania, w tym paradygmatu proceduralnego oraz obiektowego. Osiągnięcia umiejętności w ocenie przydatności paradygmatów do rozwiązywania różnego typu problemów projektowania, implementacji, testowania i debugowania prostych programów proceduralnych.

Cel 2 Poznanie składni języka C jako przykładu języka strukturalnego pozwalającego na programowanie proceduralne

Cel 3 Podstawy proceduralnego podejścia do programowania - tworzenie funkcji oraz procedur, które potrafią obsługiwać dowolne typy danych.

Cel 4 Zamykanie danych w oddzielnych plikach, udostępnianie działań nad danymi przez funkcje - elementy stylu obiektowego w języku C.

4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1 zaliczenie przedmiotu: Wstęp do programowania

5 EFEKTY KSZTAŁCENIA

EK1 Wiedza Student rozumie pojęcie paradygmat programowania. Ma wiedzę ogólną w zakresie języków i paradygmatów programowania, programowania obiektowego. Potrafi wybrać paradygmat właściwy dla problemu, który rozwiązuje oraz zna środowiska (języki) umożliwiające implementację rozwiązania problemu

EK2 Wiedza Student zna składnię C, rozumie na czym polega proceduralne podejście do programowania. Ma szczegółową wiedzę nt. algorytmiki, projektowania i programowania obiektowego.

EK3 Umiejętności Student potrafi czytać ze zrozumieniem programy napisane w C, potrafi napisać i uruchomić własny program w C, który rozwiązuje postawiony przed nim problem.

EK4 Umiejętności Umie stosować styl obiektowy przy tworzeniu prostych programów w języku C przy ożyciu struktur i unii, potrafi rozdzielić obsługę tablic, kolejek, stosów od obsługi danych.

6 TREŚCI PROGRAMOWE

LABORATORIUM		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
L1	Wprowadzenie danych z klawiatury i wyprowadzenie na monitor. Formatowanie, tworzenie tabeli.	1
L2	Operacje arytmetyczne, priorytet operacji, konwersja typów.	1
L3	Obsługa wierszy tekstowych: wprowadzenie danych z monitora, kopiowanie, sklejanie, sortowanie, poszukiwanie kontekstu.	2
L4	Tablice o dwóch indeksach przy dynamicznym alokowaniu pamięci. Zmiana rozmiaru wiersza, wymiana wierszy pomiędzy sobą, dodawanie nowego wiersza.	2
L5	Tablica obiektów struktur przy dynamicznym alokowaniu pamięci. Obsługa tablicy, obsługa obiektu.	1
L6	Wskaźniki do funkcji: program obliczający całkę oznaczoną z gładkiej funkcji jednej zmiennej rzeczywistej.	1
L7	Stworzenie systemu zapisu do pliku binarnego tablicy obiektów. Różne rekordy mają różną długość. Odczyt z pliku binarnego rekordów w dowolnej kolejności.	2

LABORATORIUM		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
L9	Hierarchia obiektów. Obsługa tablicy obiektów Triangle, zawierających obiekt Point (wierzchołek trójkąta). Generalna zasada: kontener (tablica) nie powinna zależeć od typu danych. Technika: stosowanie wskaźników void * oraz wskaźników do funkcji.	2
L10	Hierarchia obiektów. Obsługa tablicy obiektów Triangle, zawierających obiekt Point (wierzchołek trójkąta). Teraz Point wymaga dynamicznego alokowania pamięci dla tablicy współrzędnych oraz jej zwolnienia.	2
L11	Tworzymy dla poprzedniego zadania menedżer pamięci, odpowiadający za alokowanie, zwolnienie, re-alokowanie w celu przechwytywania ucieczek pamięci.	1
L12	Techniki obsługi macierzy i wektorów. Rozwijanie petli dla algorytmu obliczenia iloczynu skalarnego dwóch wektorów, funkcji pomiaru czasu.	2
L13	Przepisanie jednego obiektu do innego. Powstające problemy i sposoby rozwiązania.	1

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Pojęcie języka strukturalnego, struktura programu napisanego w języku C, pliki źródłowe i nagłówkowe. Struktura biblioteki CRT. Pojęcie o IDE. Pojęcie funkcji, funkcji statyczne, strumieni stdin, stdout, znaki specjalne (escape sequence), typy i formaty zmiennych, dyrektywy preprocesora.	1
W2	Operatory, wyrażenia, instrukcje, operator przypisania. Biblioteka funkcji matematycznych. Konwersja typów, instrukcja warunkowa, operatory arytmetyczne, relacyjne, logiczne, priorytety operatorów.	1
W3	Definicja a deklaracja funkcji, lista argumentów, argumenty formalne i faktyczne, sposoby przekazywania argumentów do funkcji, wartość zwracana przez funkcję.	1
W4	Instrukcje petli for, while, do-while, instrukcja goto. Pojęcie o wskaźnikach. Konwersje typów - jawne i niejawne	1
W5	Otwieranie i zamykanie pliku. Zapis i odczyt sformatowany. Pliki binarne. Sterowanie wskaźnikiem pozycji pliku. Zapis i odczyt do/z pliku binarnego. Tworzenie i usunięcie katalogu. Struktura pliku z rekordami o różnej długości.	2
W6	Stałe i zmienne znakowe. Tablice typu char i wchar_t. Stałe tekstowe. Funkcje CRT obsługi tablic. Wskaźniki do tablic. Dynamiczne alokowanie i zwolnienie pamięci dla tablic jednowymiarowych.	1
W7	Poruszanie się po tablicy za pomocą wskaźnika. Zmienna enum. Tablice w argumentach funkcji.	1

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W8	Statyczne wydzielenie pamięci i inicjalizacja. Dynamiczne alokowanie i zwolnienie pamięci. Pojęcie o fragmentowaniu przestrzeni adresowej i sposoby kompresji heap. Kopiowanie tablic na przekrywających się obszarach pamięci oraz inne operacji nad tablicami. Funkcji CRT obsługi heap.	2
W9	Tablice o dwóch indeksach. Inicjalizacja tablic. Pojęcie "wskaznik do wskaznika". Dynamiczne alokowanie i zwolnienie pamięci dla tablic dwuwymiarowych.	2
W10	Struktura programu w C. Zmienne lokalne, globalne, statyczne. Klasy pamięci i zakres deklaracji (obszar widoczności).	1
W11	Pojęcie struktury jako łączenie danych w całość. Obiekty struktur oraz tablice struktur. Dynamiczne alokowanie i zwolnienie pamięci.	2
W12	Wektory i macierzy. Podstawowe algorytmy algebry liniowej i osobliwości ich implementacji w języku C. Rozwijanie petli i blokowanie rejestrów.	1
W14	Struktury danych: kolejka, stos, lista. Rozdzielenie obsługi kontenera od obsługi danych. Wskazniki void * oraz wskazniki do funkcji. Tablice wskazników do funkcji.	1
W15	Bitowe operacji logiczne. Maski bitowe.	1

7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Konsultacje

N4 Zadanie projektowe

8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
Godziny kontaktowe z nauczycielem akademickim, w tym:	
Godziny wynikające z planu studiów	36
Konsultacje przedmiotowe	5
Egzaminy i zaliczenia w sesji	0
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	49
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	30
SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA	120
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	4.00

9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Odpowiedź ustna

F2 Projekt indywidualny

F3 Kolokwium

OCENA PODSUMOWUJĄCA

P1 Średnia ważona ocen formujących

WARUNKI ZALICZENIA PRZEDMIOTU

W1 Zaliczenie wszystkich kolokwium

W2 Zaliczenie projektu indywidualnego

KRYTERIA OCENY

EFEKT KSZTAŁCENIA 1	
NA OCENĘ 2.0	Student nie wie, co to jest paradygmat programowania, nie zna podstawowych paradygmatów, nie zna ważnych języków programowania, nie wie jakie paradygmaty można zrealizować w danym języku, nie potrafi dobrać właściwego paradygmatu i języka do rozwiązania postawionego przed nim zadania

NA OCENĘ 3.0	Student zna zasady proceduralnego programowania i potrafi stworzyć prosty program w języku C.
NA OCENĘ 3.5	Student zna zasady proceduralnego programowania i potrafi stworzyć złożony program w języku C, obsłużyć We/Wy do/z pliku binarnego, opracować w programie osobne systemy obsługi błędów i komunikatów.
NA OCENĘ 4.0	Student potrafi - oprócz programowania imperatywnego i obiektowego - programować w logice i zna języki umożliwiające takie podejście do programowania.
NA OCENĘ 4.5	Student zna zasady proceduralnego programowania i potrafi stworzyć program w języku C, używając stylu programowania "obektowego", potrafi stworzyć obsługę dla obiektów o strukturze zagnieżdżonej.
NA OCENĘ 5.0	Student zna zasady proceduralnego programowania i potrafi stworzyć program w języku C, używając stylu programowania "obektowego", potrafi stworzyć obsługę dla obiektów o strukturze zagnieżdżonej, a również rozdzielić funkcjonalności różnych obiektów.
EFEKT KSZTAŁCENIA 2	
NA OCENĘ 2.0	Student nie zna wielu podstawowych elementów składni języka C/C++, nie rozumie na czym polega programowanie obiektowe
NA OCENĘ 3.0	Student nie zna wielu elementów składni, ale potrafi je zastąpić innymi; potrafi napisać prosty program w wersji proceduralnej, ma kłopoty ze zrozumieniem działania programów w C na podstawie analizy kodu źródłowego napisanego przez innych programistów.
NA OCENĘ 3.5	Student opanował składnię C w stopniu pozwalającym na zwarty zapis kodu prostych programów, potrafi analizować kod prostych programów, nie potrafi napisać programu w stylu obiektowym.
NA OCENĘ 4.0	Student potrafi napisać definicje struktury i stworzyć tablice struktur.
NA OCENĘ 4.5	Student zna wszystkie elementy składni C, potrafi zaprojektować program używając styl obiektowy, umie łączyć dane w struktury oraz panuje arytmetyka wskaźników.
NA OCENĘ 5.0	Student zna składnię C w stopniu pozwalającym na tworzenie zwartych, szybkich i poprawnie działających programów, potrafi łączyć dane w struktury, dynamicznie alokować i zwalniać pamięć dla tablic obiektów dowolnego typu, umie rozdzielić funkcjonalność kontenera od obsługi danych.
EFEKT KSZTAŁCENIA 3	
NA OCENĘ 2.0	Student nie potrafi analizować kodu źródłowego napisanego przez innych
NA OCENĘ 3.0	Student rozumie działanie programu na podstawie lektury kodu źródeł.
NA OCENĘ 3.5	Potrafi napisać program w C realizujący przyjęte założenia.
NA OCENĘ 4.0	Potrafi zrobić analizę pozwalającą ustalić wymagania jakie stawiamy programowi
NA OCENĘ 4.5	Potrafi zastosować kod używający dynamicznych struktur danych

NA OCENĘ 5.0	Potrafi napisać kod optymalny pod względem wydajności i uzasadnić jego optymalność
EFEKT KSZTAŁCENIA 4	
NA OCENĘ 2.0	Student nie potrafi napisać definicji struktury
NA OCENĘ 3.0	Potrafi napisać definicję prostej struktury bez odniesienia się do potrzeb programu
NA OCENĘ 3.5	Student umie zaprojektować strukturę, unie, enumeration dopasowane do potrzeb programu
NA OCENĘ 4.0	Potrafi tworzyć własne biblioteki funkcji
NA OCENĘ 4.5	Potrafi zrobić projekt współdziałania wszystkich struktur i, unij współdziałających w programie, potrafi zamykać dane i funkcje lokalne w pliku i udostępniać działania nad danymi poprzez tworzenie funkcji interfejsowych.
NA OCENĘ 5.0	Student potrafi zaprojektować struktury, unie oraz ich obsługę w stylu obiektowym, które modelują rzeczywiste złożone obiekty, umie definiować struktury, elementami których występują obiekty innych struktur.

10 MACIERZ REALIZACJI PRZEDMIOTU

EFEKT KSZTAŁCENIA	ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	NARZĘDZIA DYDAKTYCZNE	SPOSOBY OCENY
EK1	I1_W01 I1_W04 I1_W06	Cel 1 Cel 2 Cel 3 Cel 4	L1 L2 L3 L4 L5 L6 L7 L9 L10 L11 L12 L13 W1 W2 W11 W14 W15	N1 N3	F2
EK2	I1_W01 I1_W04 I1_W06	Cel 2 Cel 3 Cel 4	L1 L2 L3 L4 L5 L6 L7 L9 L10 L11 L12 L13 W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W14 W15	N1 N2 N3	F1 F3

EFEKT KSZTAŁCENIA	ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	NARZĘDZIA DYDAKTYCZNE	SPOSOBY OCENY
EK3	I1_U01b I1_U07b I1_U10	Cel 2	L1 L2 L3 L4 L5 L6 L7 L9 L10 L11 L12 L13 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W14 W15	N1 N2 N3	F1 F3
EK4	I1_U01b I1_U03 I1_U10	Cel 3 Cel 4	L9 L10 L11 L12 L13 W1 W2 W3 W11 W12 W14 W15	N1 N2 N3	F1

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA

- [1] | B.Kernighan, D.Ritchie — *Język ANSI C*, Warszawa, 2002, WNT
- [2] | J. Richter, C. Nasarre. — *Windows via C/C++*, Microsoft Press, 2008, APN PROMISE

LITERATURA UZUPEŁNIAJĄCA

- [1] | B.Eckel — *Thinking in C++*, Gliwice, 2002, Helion
- [2] | — *MSDN - Microsoft Developer Network*, , 2020,

12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr hab. inż. prof.PK. Sergiej Fialko (kontakt: sfialko@pk.edu.pl)

OSOBY PROWADZĄCE PRZEDMIOT

- 1 dr hab. inż. Siergiy Fialko (kontakt: sfialko@pk.edu.pl)
- 2 mgr. Marcin Wątopek (kontakt: marcin.watopek@pk.edu.pl)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)

(odpowiedzialny za przedmiot)

(dziekan)



PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....

.....