Politechnika Krakowska im. Tadeusza Kościuszki

Karta przedmiotu

obowiązuje studentów rozpoczynających studia w roku akademickim 2016/2017

Wydział Fizyki, Matematyki i Informatyki

Kierunek studiów: Informatyka

Profil: Ogólnoakademicki

Forma sudiów: stacjonarne

Kod kierunku: I

Stopień studiów: I

Specjalności: Brak specjalności

1 Informacje o przedmiocie

Nazwa przedmiotu	Języki i paradygmaty programowania 1
Nazwa przedmiotu w języku angielskim	
Kod przedmiotu	WFMiI I oIS C4 16/17
Kategoria przedmiotu	Przedmioty kierunkowe
Liczba punktów ECTS	4.00
Semestry	2

2 Rodzaj zajęć, liczba godzin w planie studiów

Semestr	Wykład	Ćwiczenia	Laboratorium	Laboratorium komputerowe	Seminarium	Projekt
2	30	0	30	0	0	0

3 Cele przedmiotu

Cel 1 Poznanie podstawowych paradygmatów programowania,w tym paradygmatu proceduralnego na przykładzie języka C.

Cel 2 Opanowanie stylu programowania "obiektowego", a również opanowanie technik oddzielania funkcjonalności rożnych obiektów, na przykład - kontenera i danych.

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, zna podstawy języka C, potrafi wybrać paradygmat właściwy dla problemu, który rozwiązuje.

EK2 Wiedza Student zna składnię języka C, ma podstawowe pojęcia łączenia danych w struktury, dynamicznego alokowania i zwolnienia pamięci, elementarnej pracy z plikami, zna strukturę biblioteki CRT.

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 Kompetencje społeczne Student umie komunikować z kolegami oraz z wykładowcom.

6 Treści programowe

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. .	2
W2	Operatory, wyrażenia, instrukcji, operator przypisania. Biblioteka funkcji matematycznych. Konwersja typów, instrukcja warunkowa, operatory arytmetyczne, relacyjne, logiczne, priorytety operatorów.	2
W3	Definicja a deklaracja funkcji, lista argumentów, argumenty formalne i faktyczne, sposoby przekazywania argumentów do funkcji, wartość zwracana przez funkcje.	2
W4	Instrukcje pętli for, while, do-while, instrukcja goto. Pojęcie o wskaźnikach.	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. Tablicy typu char i wchar_t. Stałe tekstowe. Funkcji CRT obsługi tablic. Wskaźniki do tablic. Dynamiczne alokowanie i zwolnienie pamięci dla tablic jednowymiarowych.	3
W7	Poruszanie się po tablice za pomocą wskaźnika. Zmienna enum. Tablicy w argumentach funkcji.	2
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.	3

Wykład		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin
W9	Tablicy o dwóch indeksach. Inicjalizacja tablic. Pojęcie "wskaźnik do wskaźnika". 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).	2
W11	Pojęcie struktury jako łączenie danych w całość. Obiekty struktur oraz tablicy struktur. Dynamiczne alokowanie i zwolnienie pamięci.	3
W12	Wektory i macierzy. Podstawowe algorytmy algebry liniowej i osobliwości ich implementacji w języku C. Rozwijanie pętli i blokowanie rejestrów.	2
W13	Struktury danych: kolejka, stos, lista. Rozdzielenie obsługi kontenera od obsługi danych. Wskaźniki void * oraz wskaźniki do funkcji. Tablicy wskaźników do funkcji.	3
W14	Bitowe operacji logiczne. Maski bitowe.	1

Laboratorium		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin
L1	Wprowadzenie danych z klawiatury i wyprowadzenie na monitor. Formatowanie, tworzenie tabeli.	2
L2	Operacje arytmetyczne, priorytet operacji, konwersja typów.	4
L3	Obsługa wierszy tekstowych: wprowadzenie danych z monitora, kopiowanie, sklejanie, sortowanie, poszukiwanie kontekstu.	2
L4	Tablicy o dwóch indeksach przy dynamicznym alokowaniu pamięci. Zmiana rozmiaru wiersza, wymiana wierszy pomiędzy sobą, dodawanie nowego wiersza.	2
L5	Tworzenie systemu obsługi błędów i komunikatów.	2
L6	Tablica obiektów struktur przy dynamicznym alokowaniu pamięci. Obsługa tablicy, obsługa obiektu.	2
L7	Stworzenie systemu zapisu do pliku binarnego tablicy obiektów, przy czym różne rekordy mają różne długości. Odczyt z pliku binarnego rekordów w dowolnej kolejności.	2
L8	Kolokwium I	2
L9	Obiekty zagnieżdżone. 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

Laboratorium		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin
L10	Obiekty zagnieżdżone. 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.	2
L12	Techniki obsługi macierzy i wektorów. Rozwijanie pętli dla algorytmu obliczenia iloczynu skalarnego dwóch wektorów, funkcji pomiaru czasu.	2
L13	Przepisanie jednego obiektu do innego. Powstające problemy i sposoby rozwiązania.	2
L14	Kolokwium II	2

7 Narzędzia dydaktyczne

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Konsultacje

N4 Zadania domowe

8 Obciążenie pracą studenta

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe z nauczycielem akademickim, w tym:	
Godziny wynikające z planu studiów	60
Konsultacje przedmiotowe	5
Egzaminy i zaliczenia w sesji	10
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	30
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	20
Sumaryczna liczba godzin dla przedmiotu wynikająca z całego nakładu pracy studenta	**125**
Sumaryczna liczba punktów ECTS dla przedmiotu	4.00

9 Sposoby oceny

Ocena formująca

F1 Odpowiedź ustna

F2 Projekt indywidualny

F3 Kolokwium

Ocena podsumowująca

P1 Na podstawie ocen formujących

Warunki zaliczenia przedmiotu

W1 Zaliczenie wszystkich projektów indywidualnych

W2 Napisanie kolokwium zaliczeniowego.

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 zna zasady proceduralnego programowania i potrafi stworzyć program w języku C, używając stylu programowania "obiektowego".
Na ocenę 4.5	Student zna zasady proceduralnego programowania i potrafi stworzyć program w języku C, używając stylu programowania "obiektowego", 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 "obiektowego", 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 strukturalne.
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 definicję struktury zawierającej podstawowe elementy opisu i wykonującego podstawowe działania na danych. Również potrafi zrealizować program w stylu "obiektowym".
Na ocenę 4.5	Student zna wszystkie elementy składni C, potrafi zaprojektować struktury z użyciem zagnieżdżania, zaimplementować obsługę różnych obiektów w oddzielnych plikach i zabezpieczyć interakcje pomiędzy tymi obiektami.
Na ocenę 5.0	Student zna składnie C w stopniu pozwalającym na tworzenie zwartych, szybkich i poprawnie działających programów, potrafi zaprojektować i dokonać obsługę obiektów w stylu 'obiektowym' oraz oddzielić funkcjonalność kontenera od 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 uzywający struktur danych wymagających dynamicznego alokowania pamięci.
Na ocenę 5.0	Potrafi napisać kod optymalny pod względem wydajności i uzasadnić jego optymalność.
Efekt kształcenia 4	
Na ocenę 3.0	Student wykonuje projekt zespołowy tylko przy pomocy kolegów.
Na ocenę 4.0	Student wykonuje projekt zespołowy bez pomocy kolegów.
Na ocenę 5.0	Student potrafi być liderem przy wykonaniu projektu zespołowego.

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_W06	Cel 1 Cel 2	W1 W2 W3 W4 W5 W6 W7 W8 L1 L2 L3 L4 L5 L6 L7 L8 L9 L14	N1 N2 N3 N4	F1 F2 F3 P1
EK2	I1_W04 I1_W06	Cel 1 Cel 2	W5 W6 W7 W8 W9 W10 W11 W12 W13 W14 L5 L6 L7 L8 L9 L10 L11 L12 L13 L14	N1 N2 N3 N4	F1 F2 F3 P1
EK3	I1_U04 I1_U07 I1_U09	Cel 1 Cel 2	W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W14 L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12 L13 L14	N1 N2 N3 N4	F1 F2 F3 P1
EK4	I1_K03 I1_K04 I1_K06 I1_K07	Cel 1 Cel 2	W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W14 L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12 L13 L14	N1 N2 N3 N4	F1 F2 F3 P1

11 Wykaz literatury

Literatura podstawowa

[1] B.Kernighan, D.Ritchie — *Język ANSI C*, Warszawa, 2002, WNT

[2] H. Schildt. — *Programowanie C++*, Warszawa, 2002, Wydawnictwo RM

Literatura uzupełniająca

[1] J. Richter, C. Nasarre — *Windows via C/C++*, Washington, 2008, Microsoft Press

[2] S.Prata — *Język C. Szkoła Programowania*, Gliwice, 2006, Helion

12 Informacje o nauczycielach akademickich

Osoba odpowiedzialna za kartę

dr hab. inż. Sergiy Fialko (kontakt: `sfialko@pk.edu.pl`)

Osoby prowadzące przedmiot

1 dr hab. inż Sergiy Fialko (kontakt: `sfialko@poczta.onet.pl`)

2 mgr inż. Jan Wojtas (kontakt: `jwojtas@pk.edu.pl`)

13 Zatwierdzenie karty przedmiotu do realizacji

(miejscowość, data) (odpowiedzialny za przedmiot) (dziekan)

Przyjmuję do realizacji (data i podpisy osób prowadzących przedmiot)

. . . . . .

. . . . . .