

POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2022/2023

Wydział Informatyki i Telekomunikacji

Kierunek studiów: Informatyka

Profil: Ogólnoakademicki

Forma studiów: stacjonarne

Kod kierunku: I

Stopień studiów: I

Specjalności: Brak specjalności

1 INFORMACJE O PRZEDMIOCIE

NAZWA PRZEDMIOTU	Języki i paradygmaty programowania II
NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM	Programming languages and paradigms
KOD PRZEDMIOTU	WiIT I oIS C11 22/23
KATEGORIA PRZEDMIOTU	Przedmioty kierunkowe
LICZBA PUNKTÓW ECTS	6.00
SEMESTRY	3

2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁAD	ĆWICZENIA	LABORATORIUM	LABORATORIUM KOMPUTERO- WE	SEMINARIUM	PROJEKT
3	30	0	30	0	0	0

3 CELE PRZEDMIOTU

Cel 1 Poznanie podstawowych paradygmatów programowania, w tym paradygmatu imperatywnego (proceduralnego) oraz obiektowego, funkcyjnego i programowania w logice. Osiągnięcia umiejętności w ocenie przydatności paradygmatów do rozwiązywania różnego typu problemów; projektowania, implementacji, testowania i debugowania prostych programów obiektowych.

Cel 2 Podstawy obiektowego podejścia do programowania - abstrakcja danych, enkapsulacja, ukrywanie implementacji. Rozszerzenia w C++ pozwalające na programowanie obiektowe.

Cel 3 Projektowanie klas, tworzenie programów obiektowych

4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1 zaliczenie przedmiotu: języki i paradygmaty programowania I

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 implementacje rozwiązania problemu

EK2 Wiedza Student zna składnię C++, rozumie na czym polega obiektowe 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 stworzyć model obiektowy prostych programów w języku C++ przy użyciu klas, dziedziczenia, wielodziedziczenia, przeciążenia funkcji i operatorów, szablonów funkcji i klas, identyfikacji typów, wyjątków.

6 TREŚCI PROGRAMOWE

LABORATORIUM		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
L1	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
L2	Hierarchia obiektów. Obsługa tablicy obiektów Triangle, zawierających obiekt Point (wierzchołek trójkąta). Teraz Point wymaga dynamicznego alokowania pamięci dla tablicy współrzędnych oraz jej zwolnienia.	2
L3	Tworzymy dla poprzedniego zadania menedżer pamięci, odpowiadający za alokowanie, zwolnienie, re-alokowanie w celu przechwytywania ucieczek pamięci.	2
L4	Techniki obsługi macierzy i wektorów. Rozwijanie petli dla algorytmu obliczenia iloczynu skalarnego dwóch wektorów, funkcji pomiaru czasu.	2
L5	Przepisanie jednego obiektu do innego. Powstające problemy i sposoby rozwiązania.	2
L6	Inicjowanie i niszczenie obiektu. Konstruktorzy i destruktorzy.	2
L7	Hierarchia klas, wprowadzenie w dziedziczenie. Funkcje inline.	2

LABORATORIUM		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
L8	Funkcje zaprzyjaźnione do klasy	2
L9	Operatory new, delete. Alokowanie i zwolnienie pamięci.	2
L10	Przekazywanie obiektów do funkcji (przez wartość, przez wskaźnik, przez referencje). Konstruktory kopii.	2
L11	Przeciążenie operatorów binarnych. Przeciążenie operatora przypisania.	2
L12	Funkcje-szablony.	2
L13	Klasa-szablon my_vect. Tworzenie projektu z wieloma plikami. Klasy obsługi komunikatów i interfejsu. Klasy danych.	2
L14	Praca z plikami tekstowymi i binarnymi. Odczyt i zapis dokumentu.	2
L15	Funkcje wirtualne, identyfikacje typu RTTI.	2

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Przegląd ważniejszych paradygmatów programowania. Przegląd języków wspierających te paradygmaty. Zagadnienia konstrukcji języków	4
W2	Używanie obiektów (danych i funkcji składowych). Projektowanie klas. Inicjalizacja obiektów klas - konstruktory. Przeładowanie konstruktorów	2
W3	Klasy. Tworzenie i niszczenie obiektów. Konstruktorzy i destruktory.	2
W4	Wprowadzenie w dziedziczenie. Specyfikatory private, protected, public. Funkcje inline. Przypisanie obiektów. Przekazywanie obiektów do funkcji.	2
W5	Zwracanie obiektu przez funkcje. Funkcje zaprzyjaźnione.	1
W6	Przeciążenie funkcji, konstruktorzy kopii, argumenty domyślne.	2
W7	Przeciążenie operatorów.	2
W8	Dynamiczne alokowanie pamięci. Operatory new, delete.	1
W9	Dziedziczenie. Specyfikatory dostępu. Wielodziedziczenie. Klasy wirtualne.	2
W10	Polimorfizm dynamiczny. Funkcje wirtualne, abstrakcyjne, klasy abstrakcyjne.	2
W11	Szablony funkcji i klas.	2
W12	Wprowadzenie w STL, Kontenery, Wejście-wyjście w C++.	2

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W13	Obsługa wyjątków. Statyczne składowe klasy. Specyfikatory const, volatile. Niepolimorficzne rzutowanie typów.	2
W14	Przykład języka obiektowego prototypowego	2
W15	Wprowadzenie do programowania funkcyjnego	2

7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Konsultacje

N4 Platformy pracy zdalnej (MSTeams etc.)

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	15
Egzaminy i zaliczenia w sesji	15
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	60
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	30
SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA	180
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	6.00

9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Odpowiedź ustna

F2 Projekt indywidualny**F3** Kolokwium**OCENA PODSUMOWUJĄCA****P1** Ocena egzaminu**WARUNKI ZALICZENIA PRZEDMIOTU****W1** Zaliczenie wszystkich kolokwium**W2** Zaliczenie wszystkich projektów indywidualnych**W3** Zaliczenie egzaminu**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 zagnieżdzonej.
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 zagnieżdzonej, 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/C++ na podstawie analizy kodu źródłowego napisanego przez innych programistów

NA OCENĘ 3.5	Student opanował składnię C/C++ w stopniu pozwalającym na zwarty zapis kodu prostych programów, potrafi analizować kod prostych programów, nie potrafi napisać programu obiektowego
NA OCENĘ 4.0	Student potrafi napisać definicję prostej klasy będącej zawierającej podstawowe elementy opisu i wykonującego podstawowe działania na danych
NA OCENĘ 4.5	Student zna wszystkie elementy składni C/C++ , potrafi zaprojektować klasy potrzebne w implementacji programu w wersji obiektowej,
NA OCENĘ 5.0	Student zna składnię C/C++ w stopniu pozwalającym na tworzenie zwartych, szybkich i poprawnie działających programów, potrafi zaprojektować i dokonać implementacji metod klas w podejściu obiektowym.
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/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 klasy
NA OCENĘ 3.0	Potrafi napisać definicję prostej klasy bez odniesienia się do potrzeb programu
NA OCENĘ 3.5	Student umie zaprojektować klasę dopasowaną do potrzeb programu
NA OCENĘ 4.0	Potrafi wykorzystać klasy biblioteczne
NA OCENĘ 4.5	Potrafi zrobić projekt współdziałania wszystkich klas współdziałających w programie
NA OCENĘ 5.0	Student potrafi zaprojektować klasę modelującą rzeczywiste obiekty i umie stosować w programie klasy napisane przez siebie oraz klasy biblioteczne.

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_W04 I1_W06 I1_W08	Cel 1 Cel 2 Cel 3	L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12 L13 L14 L15 W1 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W14 W15	N1 N3	F2
EK2	I1_W04 I1_W06 I1_W08	Cel 2 Cel 3	L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12 L13 L14 L15 W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W14 W15	N1 N2 N3	F1 F3
EK3	I1_W04 I1_W06 I1_W08	Cel 2 Cel 3	L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12 L13 L14 L15 W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W14	N1 N2 N3	F1 F3
EK4	I1_W04 I1_W06 I1_W08 I1_W11	Cel 2 Cel 3	L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12 L13 L14 L15 W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W14	N1 N2 N3	F1

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA

[2] | J.Grębosz — *Symfonia C++ Standard*, Kraków, 2006, Editions 2000

LITERATURA UZUPEŁNIAJĄCA

[1] | B.Eckel — *Thinking in C++*, Gliwice, 2002, Helion

12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr inż. Piotr Poznański (kontakt: ppoznanski@pk.edu.pl)

OSOBY PROWADZĄCE PRZEDMIOT

1 dr inż. Piotr Poznański (kontakt: poznan@pk.edu.pl)

3 mgr inż. Łukasz Gaża (kontakt: lukasz.gaza@pk.edu.pl)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)

(odpowiedzialny za przedmiot)

(dziekan)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....

.....