

POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2021/2022

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 II
NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM	programming languages and paradigms II
KOD PRZEDMIOTU	WiIT I oIN C12 21/22
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	18	0	18	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 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). Point wymaga dynamicznego alokowania pamięci dla tablicy współrzędnych oraz jej zwolnienia.	2
L3	Tworzenie dla poprzedniego zadania menedżer pamięci, odpowiadający za 1 alokowanie, zwolnienie, re-alokowanie w celu przechwytywania ucieczek pamięci.	2
L4	Techniki obsługi macierzy i wektorów. Rozwijanie petli dla algorytmu obliczenia 1 iloczynu skalarnego dwóch wektorów, funkcji pomiaru czasu.	1
L5	Przepisanie jednego obiektu do innego. Powstające problemy i sposoby rozwiązania.	1
L6	Inicjowanie i niszczenie obiektu. Konstruktorzy i destruktory.	1
L7	Hierarchia klas, wprowadzenie w dziedziczenie. Funkcje inline.	1

LABORATORIUM		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
L8	Funkcji zaprzyjaźnione do klasy	1
L9	Operatory new, delete. Alokowanie i zwolnienie pamięci.	1
L10	Przekazywanie obiektów do funkcji (przez wartość, przez wskaźnik, przez 1 referencje). Konstruktory kopii.	1
L11	Przeciążenie operatorów binarnych. Przeciążenie operatora przypisania.	1
L12	Funkcje-szablony.	1
L13	Klasa-szablon my_vect. Tworzenie projektu z wieloma plikami. Klasy obsługi 2 komunikatów i interfejsu. Klasy danych.	1
L14	Praca z plikami tekstowymi i binarnymi. Odczyt i zapis dokumentu.	1
L15	Funkcje wirtualne, identyfikacje typu RTTI.	1

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Przegląd paradygmatów i języków. Paradygmat obiektowy. Mechanizmy bezpieczeństwa języków programowania (typizacja itd.)	2
W2	Rozszerzenia względem C (nowe mechanizmy, operatory, klasy)	2
W3	Klasy. Tworzenie i niszczenie obiektów. Konstruktorzy i destruktorzy.	2
W4	Wprowadzenie w dziedziczenie. Specyfikatory private, protected, public. Funkcje inline. Przypisanie obiektów. Przekazywanie obiektów do funkcji. Funkcje zaprzyjaźnione	1
W5	Przeciążenie funkcji, konstruktorzy kopii, argumenty domyślne.	1
W6	Przeciążenie operatorów.	1
W7	Dynamiczne alokowanie pamięci. Operatory new, delete.	2
W8	Dziedziczenie. Specyfikatory dostępu. Wielodziedziczenie.	1
W9	Polimorfizm dynamiczny. Funkcje wirtualne, abstrakcyjne, klasy abstrakcyjne. Identyfikacje typu na etapie wykonania (RTTI) .	2
W10	Szablony funkcji i klas.	1
W11	Przykład obiektowego języka programowania opartego o model prototypowy	1

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W12	Obsługa wyjątków. Statyczne składowe klasy. Specyfikatory const, volatile. Niepolimorficzne rzutowanie typów.	2

7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Konsultacje

N4 Narzędzia pracy zdalnej

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	15
Egzaminy i zaliczenia w sesji	6
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	63
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	60
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 Ocena projektu laboratoryjnego

F3 Kolokwium

OCENA PODSUMOWUJĄCA**P1** Ocena z egzaminu**WARUNKI ZALICZENIA PRZEDMIOTU****W1** Zaliczenie wszystkich projektów laboratoryjnych**W2** 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 wie, co to jest paradygmat programowania, zna podstawowe paradygmy
NA OCENĘ 3.5	Student wie, co to jest paradygmat programowania, zna podstawowe paradygmaty, zna ważne języków programowania,
NA OCENĘ 4.0	Student wie, co to jest paradygmat programowania, zna podstawowe paradygmaty, zna ważne języków programowania, wie jakie paradygmaty można zrealizować w danym języku,
NA OCENĘ 4.5	Student wie, co to jest paradygmat programowania, zna podstawowe paradygmaty, zna ważne języków programowania, wie jakie paradygmaty można zrealizować w danym języku, potrafi dobrać właściwy paradygmat
NA OCENĘ 5.0	Student wie, co to jest paradygmat programowania, zna podstawowe paradygmaty, zna ważne języków programowania, wie jakie paradygmaty można zrealizować w danym języku, potrafi dobrać właściwy paradygmat i język do rozwiązania postawionego przed nim zadania
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_W11 I1_U09	Cel 1 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	N1 N2 N3 N4	F1 F2 F3 P1
EK2	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 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12	N1 N2 N3 N4	F1 F2 F3 P1
EK3	I1_W04 I1_W06	Cel 1 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	N1 N2 N3 N4	F1 F2 F3 P1
EK4	I1_W04 I1_W06	Cel 1 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	N1 N2 N3 N4	F1 F2 F3 P1

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA

[1] 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: ppoznanski@pk.edu.pl)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)

(odpowiedzialny za przedmiot)

(dziekan)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....