

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: 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
KOD PRZEDMIOTU	WiIT I oIN 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	18	0	18	0	0	0

3 CELE PRZEDMIOTU

Cel 1 Poznanie podstawowych paradygmatów programowania obiektowego - enkapsulacja, dziedziczenie, polimorfizm. Osiągnięcia umiejętności w rozwiązywaniu różnego typu problemów; projektowania, implementacji, testowania i debugowania prostych programów obiektowych.

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

Kod archiwizacji:

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 implementację 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

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BŁOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Paradygmat obiektowy: enkapsulacja, dziedziczenie, polimorfizm. Przestrzeni nazw. Wprowadzenia do klas. Konstruktory i destruktory. Konstruktory kopiujące.	2
W2	Wprowadzenie w dziedziczenie. Specyfikatory private, protected, public. Funkcje inline.	1
W3	Przekazywanie obiektów do funkcji. Zwracanie obiektów przez funkcje. Funkcje zaprzyjaźnione. Przeciążenie funkcji. Argumenty domyślne.	3
W4	Przeciążenie operatorów.	2
W5	Dynamiczne alokowanie pamięci. Operatory new, delete. Dziedziczenie i wielodziedziczenie. Klasy wirtualne.	1
W6	Polimorfizm dynamiczny. Funkcje wirtualne, abstrakcyjne. Klasy abstrakcyjne.	2
W7	Identyfikacja typu na etapie wykonania (RTTI).	2
W8	Szablony funkcji i klas.	2
W11	Wejście-wyjście w C++. Obsługa wyjątków. Statyczne składowe klasy. Specyfikatory const, volatile. Niepolimorficzne rzutowanie typów.	2
W12	Wprowadzenie do STL.	1

LABORATORIUM		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
L1	Wprowadzenie w C++. Inkapsulacja. Pojęcia klasy - dane i metody.	2
L2	Inicjowanie i niszczenie obiektu. Konstruktory i destruktory. Hierarchia klas, wprowadzenie w dziedziczenie.	2
L3	Konstruktory kopiujące. Funkcje inline. Funkcje zaprzyjaźnione.	2
L4	Operatory new, delete. Obsługa pamięci dynamicznej.	2
L5	Przeciążenie operatorów binarnych. Przeciążenie operatora przypisania.	2
L6	Przekazywanie obiektów do funkcji (przez wartość, przez wskaźnik, przez referencje).	1
L7	Funkcje - szablony, klasy-szablony.	2
L8	Praca z plikami tekstowymi i binarnymi. Odczyt i zapis dokumentu.	2
L9	Funkcje wirtualne, destruktory wirtualne, identyfikacja typu (RTTI)	2
L10	Elementy STL.	1

7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Konsultacje

N4 Narzędzia pracy zdalnej MS TEAMS

N5 strona <http://torus.uck.pk.edu.pl/fialko> (strona Sergiy Fialko)

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	45
Egzaminy i zaliczenia w sesji	9
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	40
Opracowanie wyników	0
Przygotowanie raportu, projektu, prezentacji, dyskusji	50
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 zna podstaw programowania obiektowego
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 podstawowych elementów składni języka C/C++, nie rozumie na czym polega enkapsulacja, polimorfizm, dziedziczenie.
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_W01 I1_W04 I1_W06	Cel 1 Cel 2 Cel 3	W1 W2 W3 W4 W5 W6 W7 W8 W11 W12 L1 L2 L3 L4 L5 L6 L7 L8 L9 L10	N1 N2 N3 N4	F1 F2 F3 P1
EK2	I1_W01 I1_W06 I1_W08	Cel 1 Cel 2 Cel 3	W1 W2 W3 W4 W5 W6 W7 W8 W11 W12 L1 L2 L3 L4 L5 L6 L7 L8 L9 L10	N1 N2 N3 N4	F1 F2 F3 P1
EK3	I1_U01b I1_U06b I1_U23	Cel 1 Cel 2 Cel 3	W1 W2 W3 W4 W5 W6 W7 W8 W11 W12 L1 L2 L3 L4 L5 L6 L7 L8 L9 L10	N1 N2 N3 N4	F1 F2 F3 P1
EK4	I1_W11 I1_U01b I1_U03	Cel 1 Cel 2 Cel 3	W1 W2 W3 W4 W5 W6 W7 W8 W11 W12 L1 L2 L3 L4 L5 L6 L7 L8 L9 L10	N1 N2 N3 N4	F1 F2 F3 P1

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA

[1] | **B.Stroustrup** — *The C++ Programming Language (Fourth Edition)*, USA, 2013, Addison Wesley

[2] | **H. Schild** — *The Complete Reference C++*, USA, 2020, Tata McGraw Hill

LITERATURA UZUPEŁNIAJĄCA

[1] | — *C++ language documentation. MSDN*, , 0,

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

2 mgr. inż. Krzysztof Swaldek (kontakt: krzysztof.swaldek@pk.edu.pl)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)

(odpowiedzialny za przedmiot)

(dziekan)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....
.....