

KARTA PRZEDMIOTU

Kod przedmiotu	0613-2INF-C22-PRO	
Nazwa przedmiotu w języku	polskim	Programowanie obiektowe Object oriented programming
	angielskim	

1. USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

1.1. Kierunek studiów	Informatyka
1.2. Forma studiów	stacjonarne
1.3. Poziom studiów	studia I-stopnia inżynierskie
1.4. Profil studiów	ogólnoakademicki
1.5. Osoba przygotowująca kartę przedmiotu	Mirosław Głowacki
1.6. Kontakt	glowacki@metal.agh.edu.pl

2. OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

2.1. Język wykładowy	polski
2.2. Wymagania wstępne	Środowisko programisty Wstęp do programowania

3. SZCZEGÓŁOWA CHARAKTERYSTYKA PRZEDMIOTU

3.1. Forma zajęć	wykłady, laboratorium	
3.2. Miejsce realizacji zajęć	Zajęcia w pomieszczeniu dydaktycznym UJK	
3.3. Forma zaliczenia zajęć	wykłady – egzamin laboratorium – zaliczenie z oceną	
3.4. Metody dydaktyczne	wykład, zajęcia laboratoryjne przy komputerach	
3.5. Wykaz literatury	podstawowa	<ol style="list-style-type: none"> Grębosz, Misja w nadprzestrzeń C++14/17, Wyd. I, Helion 2020. J. Grębosz, Opus magnum C++11, Wyd. I (3 tomy), Helion 2017 B. Stroustrup, Język C++. Kompendium wiedzy, Wyd. IV, Helion 2014 J. Grębosz, Symfonia C++, Oficyna Kallimach, Kraków 2006 J. Grębosz, Pasja C++, Oficyna Kallimach, Kraków
	uzupełniająca	<ol style="list-style-type: none"> S. Prata. Język C++. Szkoła programowania. Wydanie VI, Helion 2013 Nowak R., Pająk A., Język C++. Mechanizmy, wzorce, biblioteki, BTC, Legionowo 2010 McLaughlin B. D., Pollice G., West D., Rusz głową. Analiza i projektowanie obiektowe, Helion, Gliwice 2010 B. Meyer, Programowanie zorientowane obiektowo, Helion, Gliwice 2005 Shalloway A., Trott J. T., Projektowanie zorientowane obiektowo. Wzorce projektowe, Helion, Gliwice 2005

4. CELE, TREŚCI I EFEKTY UCZENIA SIĘ

4.1. Cele przedmiotu

Wykład:

- C1. Celem wykładu jest zapoznanie studentów z paradygmatem programowania obiektowo zorientowanego. Studenci poznają programowanie zgodne z postrzeganiem rzeczywistości, kryteria obiektowości oraz techniki programowania obiektowego. Wykład poświęcony jest zarówno obiektowemu modelowaniu dziedziny jak i pojęciom oraz ogólnym zasadom projektowania i programowania obiektowego.
- C2. Równolegle przedstawiane są mechanizmy programistyczne umożliwiające implementację modelu obiektowego w oparciu o język C++. W trakcie kolejnych wykładów omawiane są: klasy, tryb dostępu do składników klas, konwersje, przeciążenia operatorów, dziedziczenie polimorfizm i generyczność.

Laboratorium:

- C1. W trakcie ćwiczeń laboratoryjnych studenci poznają aspekty programowania obiektowo zorientowanego od strony praktycznej wykonując mini-projekty zlecane przez prowadzących w trakcie zajęć laboratoryjnych. W ramach prac własnych realizują końcową wersję implementacji projektu bazując na informacjach uzyskanych w trakcie wykładu i ćwiczeń laboratoryjnych oraz przygotowują sprawozdania.

4.2. Treści programowe

Wykład

Przegląd podstawowych paradygmatów programowania. Środowiska programistyczne zorientowane obiektowo. Pojęcie obiektu, proste przykłady obiektów, analogia do obiektów rzeczywistych. Obiektowe modelowanie dziedziny. Cechy programowania obiektowego. Typy obiektów, typy definiowane przez użytkownika, statyczna kontrola typów. Cechy obiektów – pola (atrybuty) i metody. Klasy i kapsułkowanie - klasy jako typy, składniki klas, wskaźnik *this*, konstruktory i destruktory, statyczne składniki klas, pola stałe i ulotne, pola wskaźnikowe i referencyjne, lista inicjalizacyjna konstruktora. Obiekty i ich konstruktory - obiekty lokalne i globalne, statyczne obiekty lokalne. Tryby dostępu do składników klas, ukrywanie informacji, funkcje zaprzyjaźnione. Przestrzenie nazw. Automatyczna konwersje typów, rzutowania typów, rodzaje rzutowań. Wskaźniki do instancji i składników klas. Operatory i ich przeciążenia. Dziedziczenie klas, dostęp do składników klas bazowych i pochodnych, dziedziczenie a zawieranie klas, ponowna definicja składników klas. Dziedziczenie wielokrotne i wielopokoleniowe - ryzyko wieloznaczności, klasy wirtualne. Niejawna konwersja typów w warunkach dziedziczenia, wiązanie dynamiczne. Funkcje wirtualne i polimorfizm, klasy abstrakcyjne. Wyjątki i ich obsługa. Generyczność - szablony funkcji i klas, parametry szablonów, klasy specjalizowane, ograniczona generyczność, Biblioteka STL.

Laboratorium

Zintegrowane środowisko programistyczne zorientowane obiektowo. Porównanie paradygmatów programowania proceduralnego i obiektowo zorientowanego w języku C++. Definiowanie klas i ich pól o dostępie publicznym. Metody klas, deklarowanie i definiowanie metod wewnątrz i na zewnątrz klasy. Tworzenie instancji klas – przykłady prostych obiektów, rola wskaźnika *this*, odwzorowanie obiektów rzeczywistych – obiekty geometryczne. Tworzenie klas ze składnikami prywatnymi, ukrywanie i wybiórcze udostępnianie informacji, dostęp do prywatnych pól klasy – metody publiczne i funkcje zaprzyjaźnione. Praca z konstruktorami i destruktorami, definiowanie własnych konstruktorów, inicjalizacja pól stałych i referencyjnych, definiowanie tablic dynamicznych – rola konstruktorów i destruktora. Korzystanie z automatycznej konwersji typów, definiowanie operatorów konwersji i konstruktorów konwertujących. Zastosowanie operatorów rzutowania typów. Stosowanie wskaźników do składników klas i ich tablic. Przeciążanie operatorów. Obiektowe modelowanie dziedziny - projektowanie i implementacja prostych programów obiektowo zorientowanego. Korzystanie z mechanizmu dziedziczenia klas, uzyskiwanie dostępu do składników klas bazowych i klas zawartych, tworzenie list inicjalizacyjnych konstruktorów klas pochodnych. Usuwanie problemów związanych z dziedziczeniem wielokrotnym, tworzenie klas wirtualnych. Używanie wskaźników do instancji klas, poznanie mechanizmu niejawnej konwersji typów i wiązania dynamicznego. Tworzenie funkcji wirtualnych, korzystanie z polimorfizm, tworzenie klas abstrakcyjnych. Budowa prostych programów generycznych – korzystanie z funkcji, klas i metod szablonowych.

4.3. Przedmiotowe efekty uczenia się

Efekt	Student, który zaliczył przedmiot	Odniesienie do kierunkowych efektów uczenia się
w zakresie WIEDZY:		
W01	zna podstawy teorii programowania obiektowego, rozumie sposób podejścia do obiektów oraz ich metod i pól, a także sposoby wzajemnego powiązania klas obiektów i komunikowania się poszczególnych obiektów	INF1A_W07 INF1A_W012 INF1A_W013
W02	definiuje hierarchię klas, przekazywanie komunikatów pomiędzy obiektami oraz zaprojektuje sprawnie działającą aplikację opartą o teorię programowania obiektowo zorientowanego	INF1A_W07 INF1A_W012 INF1A_W013
w zakresie UMIEJĘTNOŚCI:		
U01	analizuje przedstawiony mu problem pod kątem wymagań programistycznych oraz zaprojektuje odpowiedni system rozwiązania zadań z niego wynikających	INF1A_U5 INF1A_U11 INF1A_U13 INF1A_U15 INF1A_U20
U02	ma zdolność do samodzielnego zaimplementowania aplikacji w języku C++, usunięcia błędów i przetestowania obiektowo zorientowanego kodu aplikacji	INF1A_U5 INF1A_U11 INF1A_U13 INF1A_U15 INF1A_U20
w zakresie KOMPETENCJI SPOŁECZNYCH:		
K01	jest zdolny do rozwiązywania konkretnych problemów z życia codziennego przy pomocy uzyskanej wiedzy i umiejętności praktycznych oraz umie zaproponować ich rozwiązanie z wykorzystaniem narzędzi informatycznych	INF1A_K01 INF1A_K02 INF1A_K06

4.4. Sposoby weryfikacji osiągnięcia przedmiotowych efektów uczenia się

Efekty przedmiotowe (symbol)	Sposób weryfikacji (+/-)														
	Egzamin pisemny			Kolokwium			Zadania domowe			Sprawozdania			Inne		
	Forma zajęć			Forma zajęć			Forma zajęć			Forma zajęć			Forma zajęć		
	W	L	P	W	W	W	W	L	P	W	L	P	W	L	P
W01	+			+											
W02	+			+											
U01								+				+			
U02								+				+			
K01	+							+				+			

4.5. Kryteria oceny stopnia osiągnięcia efektów uczenia się		
Forma zajęć	Ocena	Kryterium oceny
wykład (W)	3	co najmniej 50% i nie więcej niż 60% łącznej liczby punktów możliwych do uzyskania
	3,5	ponad 60% i nie więcej niż 70% łącznej liczby punktów możliwych do uzyskania
	4	ponad 70% i nie więcej niż 80% łącznej liczby punktów możliwych do uzyskania
	4,5	ponad 80% i nie więcej niż 90% łącznej liczby punktów możliwych do uzyskania
	5	ponad 90% liczby punktów możliwych do uzyskania
laboratorium (L)	3	co najmniej 50% i nie więcej niż 60% łącznej liczby punktów możliwych do uzyskania
	3,5	ponad 60% i nie więcej niż 70% łącznej liczby punktów możliwych do uzyskania
	4	ponad 70% i nie więcej niż 80% łącznej liczby punktów możliwych do uzyskania
	4,5	ponad 80% i nie więcej niż 90% łącznej liczby punktów możliwych do uzyskania
	5	ponad 90% liczby punktów możliwych do uzyskania
projekt (P)	3	co najmniej 50% i nie więcej niż 60% łącznej liczby punktów możliwych do uzyskania
	3,5	ponad 60% i nie więcej niż 70% łącznej liczby punktów możliwych do uzyskania
	4	ponad 70% i nie więcej niż 80% łącznej liczby punktów możliwych do uzyskania
	4,5	ponad 80% i nie więcej niż 90% łącznej liczby punktów możliwych do uzyskania
	5	ponad 90% liczby punktów możliwych do uzyskania

5. BILANS PUNKTÓW ECTS - NAKŁAD PRACY STUDENTA

Kategoria	Obciążenie studenta	
	Studia stacjonarne	Studia niestacjonarne
LICZBA GODZIN REALIZOWANYCH PRZY BEZPOŚREDNIM UDZIALE NAUCZYCIELA /GODZINY KONTAKTOWE/	90	
Udział w wykładach*	30	
Udział w laboratoriach*	30	
Realizacja projektu	30	
SAMODZIELNA PRACA STUDENTA /GODZINY NIEKONTAKTOWE/	85	
Przygotowanie do wykładu*	10	
Przygotowanie do laboratorium*	45	
Przygotowanie do egzaminu/kolokwium*	20	
Zebrań materiałów do projektu, kwerenda internetowa*	10	
ŁĄCZNA LICZBA GODZIN	175	
PUNKTY ECTS za przedmiot	7	

*niepotrzebne usunąć

Przyjmuję do realizacji (data i podpisy osób prowadzących przedmiot w danym roku akademickim)

29 listopada 2024

Mirosław Gładki