**Nazwa przedmiotu:**

Programowanie 2 - obiektowe

**Koordynator przedmiotu:**

dr inż. Ewa Pawelec

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka i Systemy Informacyjne

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

1120-IN000-ISP-0125

**Semestr nominalny:**

2 / rok ak. 2023/2024

**Liczba punktów ECTS:**

5

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1. godziny kontaktowe – 65 h; w tym
 a) obecność na wykładach – 30 h
 b) obecność na laboratoriach – 30 h
 c) konsultacje – 5 h
2. praca własna studenta – 65 h; w tym
 a) zapoznanie się z literaturą – 15 h
 b) przygotowanie się do zajęć laboratoryjnych – 50 h
Razem 130 h, co odpowiada 5 pkt. ECTS

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

1. obecność na wykładach – 30 h
2. obecność na laboratoriach – 30 h
3. konsultacje – 5 h
Razem 65 h, co odpowiada 2 pkt. ECTS

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

1. obecność na laboratoriach – 30 h
2. przygotowanie do zajęć laboratoryjnych – 50 h
Razem 80 h, co odpowiada 3 pkt. ECTS

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 30h |
| Projekt:  | 0h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Programowanie 1 – strukturalne

**Limit liczby studentów:**

Laboratoria (ćwiczenia komputerowe) – 15 os. /grupa

**Cel przedmiotu:**

Celem przedmiotu jest wyrobienie umiejętności posługiwania się podstawowymi pojęciami i konstrukcjami obiektowego języka programowania. Po ukończeniu kursu studenci powinni:
- posiadać podstawową wiedzę na temat pojęć objętych paradygmatem programowania obiektowego,
- umieć samodzielnie definiować typy danych opisujące proste obiekty odwzorowujące podstawowe pojęcia matematyczne (np. liczby zespolone, wektory, macierze),
- umieć samodzielnie definiować typy danych opisujące bardziej złożone obiekty odwzorowujące abstrakcyjne dynamiczne struktury danych (np. lista, kolejka, stos, drzewo),
- umieć wykorzystywać pliki (strumienie) w samodzielnie realizowanych programach,
- umieć samodzielnie realizować proste aplikacje konsolowe w postaci projektów jedno- i wieloplikowych,
- umieć wykorzystywać elementy biblioteki standardowej języka STL (typy wzorcowe i algorytmy) w samodzielnie realizowanych programach,
- umieć pracować w profesjonalnym środowisku uruchomieniowym (MS Visual Studio) i wykorzystywać je do pisania, testowania, debuggowania i tworzenia wersji wynikowej aplikacji.

**Treści kształcenia:**

Programowanie obiektowe - podstawowe cechy, paradygmaty programowania obiektowego.
Specyficzne elementy języka C++: strumieniowe operacje I/O; referencje; przeciążanie operatorów i funkcji; dynamiczne zarządzanie pamięcią.
Definiowanie struktur i prostych klas. Zasięg składowych klasy.
Tworzenie, usuwanie i inicjalizowanie obiektów.
Własności i atrybuty metod składowych. Funkcje zaprzyjaźnione, zaprzyjaźnianie klas i komunikacja klas. Definiowanie i przeciążanie operatorów. Przeciążanie funkcji. Tablice obiektów.
Klasa string.
Operatory new, delete.
Klasa z polem dynamicznym. Klasa z polem obiektowym.
Dziedziczenie. Rodzaje dziedziczenia.
Polimorfizm i metody wirtualne. Klasy abstrakcyjne.
Obsługa błędów. Obsługa wyjątków.
Funktory. Konwersje, operatory konwersji i konwertery.
Strumienie. Operacje na plikach tekstowych oraz binarnych.
Definiowanie i wykorzystanie szablonów funkcji.
Definiowanie i wykorzystanie szablonów klas.
Elementy biblioteki STL: podstawowe kontenery, iteratory, algorytmy.
Wybrane elementy nowych standardów C++, m.in. wyrażenia lambda, semantyka przenoszenia.

**Metody oceny:**

Zaliczenie przedmiotu na podstawie laboratorium. Obecność na zajęciach jest obowiązkowa.
Zajęcia laboratoryjne obejmują wykonanie 13 zadań o różnym stopniu trudności, na każdych zajęciach zadanie jest punktowane oddzielnie. Nie przewiduje się możliwości poprawiania ocen z poszczególnych zadań. Na ostatnich zajęciach można poprawić jedno najgorzej ocenione zadanie (szczegóły w regulaminie przedmiotu).
Ostateczna ocena zależy od sumy uzyskanych punktów: 51%-60% - 3; 61%-70% - 3,5; 71%-80% - 4; 81%-90% - 4,5; 91%-100% - 5.

**Egzamin:**

nie

**Literatura:**

1. J. Grębosz, Opus Magnum C++11. Programowanie w C++, Helion 2018 (wcześniej: Symfonia C++ standard).
2. S. Prata, Język C++. Szkoła programowania, Helion, Gliwice, 2006.
3. W. Savitch, Programowanie w tonacji C++, RM, Warszawa, 2005.
4. B. Stroustrup B., Język C++ kompendium wiedzy, Helion, Warszawa, 2014.
5. B. Eckel, Thinking in C++. Edycja polska, Helion, Warszawa, 2002.
6. N.M. Josuttis, C++ biblioteka standardowa (wyd.II), Helion, Warszawa, 2014.
7. N. Wirth, Algorytmy + struktury danych = programy, WNT, Warszawa, 2004.
8. Dokumentacja: http://en.cppreference.com
9. Materiały z wykładów: http://www.mini.pw.edu.pl/~epawelec

**Witryna www przedmiotu:**

**Uwagi:**

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka W01:**

Zna podstawowe pojęcia związane z paradygmatem programowania obiektowego i ich szczegółową egzemplifikację w wybranym języku programowania (C++)

Weryfikacja:

punktowane zadania laboratoryjne

**Powiązane charakterystyki kierunkowe:** K\_W06, K\_W08

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W02:**

Zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych realizowanych w języku C++

Weryfikacja:

punktowane zadania laboratoryjne

**Powiązane charakterystyki kierunkowe:** K\_W04, K\_W10, K\_W12, K\_W13

**Powiązane charakterystyki obszarowe:**

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U01:**

Potrafi – zgodnie z zadaną specyfikacją – zaprojektować oraz zrealizować prostą aplikację informatyczną, używając samodzielnie definiowanych typów danych i elementów biblioteki standardowej, w tym strumieni

Weryfikacja:

punktowane zadania laboratoryjne

**Powiązane charakterystyki kierunkowe:** K\_U12, K\_U30

**Powiązane charakterystyki obszarowe:**

**Charakterystyka U02:**

Ma umiejętność pracy w profesjonalnym środowisku uruchomieniowym (MS Visual Studio) , a w szczególności potrafi korzystać z dostępnych w nim narzędzi do przeprowadzania testów funkcjonalnych opracowanych aplikacji

Weryfikacja:

punktowane zadania laboratoryjne

**Powiązane charakterystyki kierunkowe:** K\_U11, K\_U21

**Powiązane charakterystyki obszarowe:**