**Nazwa przedmiotu:**

Programowanie 2 - obiektowe

**Koordynator przedmiotu:**

dr inż. Ewa Pawelec

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

1120-IN000-ISP-0125

**Semestr nominalny:**

2 / rok ak. 2017/2018

**Liczba punktów ECTS:**

5

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

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

**Język prowadzenia zajęć:**

polski

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

**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 użytkowe w postaci projektów jedno- i wieloplikowych,
- umieć wykorzystywać elementy biblioteki standardowej języka (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.
Specyficzne elementy języka C++: strumieniowe operacje I/O; przekazywanie argumentów przez referencje, typy referencyjne; argumenty domyślne; przeciążanie operatorów i funkcji; dynamiczne zarządzanie pamięcią.
Definiowanie struktur i prostych klas. Konstruktory, destruktor. Zasięg składowych klasy.
Tworzenie, usuwanie i inicjalizowanie obiektów.
Własności metod składowych. Atrybuty: static, const. Wskaźnik this. Funkcje zaprzyjaźnione. Definiowanie i przeciążanie operatorów. Przeciążanie funkcji. Tablice obiektów.
Operatory new, delete.
Klasa z polem alokowanym dynamicznie. Klasa z polem obiektowym.
Dziedziczenie, dziedziczenie wielobazowe.
Polimorfizm i metody wirtualne. Metody czysto wirtualne i abstrakcyjny typ danych (ATD).
Obsługa błędów. Obsługa wyjątków.
Funktory. Konwersje, operatory konwersji i konwertery. Strumienie. Operacje na plikach.
Definiowanie i wykorzystanie szablonów funkcji. Definiowanie i wykorzystanie szablonów klas. Klasa string. Elementy biblioteki STL (Standard Template Library): kontenery, iteratory, algorytmy.
Wybrane elementy standardu C++11. C++ a inne języki programowania obiektowego.

**Metody oceny:**

Zaliczenie przedmiotu wymaga uzyskania zaliczenia zajęć laboratoryjnych. Obecność na zajęciach jest obowiązkowa.
Zajęcia laboratoryjne obejmują wykonanie 13 zadań o różnym stopniu trudności, punktowanych w skali od 1 do 5 (maksymalnie). Ocena z laboratorium (ocena pracy semestralnej) jest ustalana przez prowadzących zajęcia na podstawie sumy ważonej ocen z wszystkich zadań. Nie przewiduje się możliwości poprawiania ocen z poszczególnych zadań.

**Egzamin:**

nie

**Literatura:**

1. B. Stroustrup, Język C++, WNT, 2002.
2. B. Eckel, Thinking in C++. Edycja polska, Helion, 2002.
3. J. Grębosz, Symfonia C++ standard, Edition, 2005.
4. N. M. Josuttis, C++ Biblioteka standardowa. Podręcznik programisty, Helion 2003.

**Witryna www przedmiotu:**

e.mini.pw.edu.pl

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt W01:**

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

Weryfikacja:

punktowane zadania laboratoryjne

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

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W04

**Efekt W02:**

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

Weryfikacja:

punktowane zadania laboratoryjne

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

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W07, T1A\_W07, T1A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt 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 efekty kierunkowe:** K\_U12, K\_U30

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U16

**Efekt 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 efekty kierunkowe:** K\_U11, K\_U21

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U14, T1A\_U15, T1A\_U09, T1A\_U13, T1A\_U14

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt K01:**

Potrafi pracować indywidualnie oraz rozumie konieczność dostosowania się do standardów i dobrych praktyk programowania

Weryfikacja:

punktowane zadania laboratoryjne

**Powiązane efekty kierunkowe:** K\_K05

**Powiązane efekty obszarowe:** T1A\_K03, T1A\_K04