**Nazwa przedmiotu:**

Programowanie

**Koordynator przedmiotu:**

dr inż. Tomasz Marks

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

**Semestr nominalny:**

2 / rok ak. 2012/2013

**Liczba punktów ECTS:**

5

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

1. godziny kontaktowe - 60 h; w tym
a. obecność na wykładach – 30 h
b. obecność na laboratoriach – 30 h
2. przygotowanie się do zajęć - 60 h; w tym
a. zapoznanie się z literaturą – 40 h
b. przygotowanie się do zajęć laboratoryjnych – 30 h
Razem nakład pracy studenta 30+30+30+30 = 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
Razem 30+30 = 60 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 – 30 h
Razem 30+30= 60 h, co odpowiada 2 pkt. ECTS

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

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

**Wymagania wstępne:**

Podstawy programowania strukturalnego

**Limit liczby studentów:**

Bez limitu

**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:**

1. Język C a C++. Definiowanie prostych klas.
2. Podstawowe założenia paradygmatu obiektowego.
3. Typy referencyjne. Domyślne wartości argumentów. Przeciążanie funkcji.
4. Konstruktory, destruktory. Definiowanie operatorów. Strumieniowe operacje we/wy.
5. Wskaźnik this. Operatory new, delete. Funkcje zaprzyjaźnione.
6. Klasa Vector z dynamicznie alokowaną tablicą składowych.
7. Pola i metody statyczne. Dostępność składowych klasy.
8. Klasa z polem obiektowym typu Vector.
9. Dziedziczenie, polimorfizm i metody wirtualne.
10. Jak realizowany jest mechanizm wirtualności? Metody czysto wirtualne i ATD.
11. Obsługa błędów. Obsługa wyjątków. Wyjątki jako mechanizm sterujący.
12. Konwersje, operatory konwersji i konwertery.
13. Definiowanie i wykorzystanie szablonów funkcji i klas.
14. Elementy biblioteki STL.
15. 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.
Uwagi dodatkowe:
Po każdym wykładzie jego treść jest udostępniana w formie pliku \*.pdf na stronie wykładowcy: www.mini.pw.edu.pl/~tmarks

**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:**

brak

**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:

kolokwium

**Powiązane efekty kierunkowe:** K\_W06

**Powiązane efekty obszarowe:** T1A\_W03

**Efekt W02:**

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

Weryfikacja:

kolokwium

**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

**Powiązane efekty kierunkowe:** K\_U30

**Powiązane efekty obszarowe:** 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

**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 zarządzać swoim czasem

Weryfikacja:

punktowane zadania

**Powiązane efekty kierunkowe:** K\_K05

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