**Nazwa przedmiotu:**

Computer Science II

**Koordynator przedmiotu:**

dr inż. Sławomir Paśko

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Mechatronics

**Grupa przedmiotów:**

Obowiązkowe

**Kod przedmiotu:**

CS2

**Semestr nominalny:**

2 / rok ak. 2020/2021

**Liczba punktów ECTS:**

5

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

1) Liczba godzin, które wymagają obecności nauczyciela - 30 godzin, w tym:
a) obecność na wykładach - 15 godzin;
b) obecność na ćwiczeniach - 15 godzin;
2) Liczba godzin samodzielnej pracy studenta - 95 godzin, w tym:
a) powtórzenie i przestudiowanie wybranych tematów kursu CSI - 5 godzin;
b) przygotowanie do testu teoretycznego - 20 godzin;
c) przygotowanie do ćwiczeń - 15 godzin;
d) przygotowanie do testu praktycznego - 55 godzin;
Suma godzin: 125 godzin = 5 ECT

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

1) Liczba godzin, w których wymagana jest obecność nauczyciela - 30 godzin, w tym
a) obecność na wykładach - 15 godzin;
b) obecność na ćwiczeniach - 15 godzin;
Suma godzin: 30 godzin = 1 ECTS

**Język prowadzenia zajęć:**

angielski

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

a) obecność na ćwiczeniach - 15 godzin;
b) przygotowanie do ćwiczeń - 15 godzin;
c) przygotowanie do testu praktycznego - 55 godzin;
Suma godzin: 85 godzin = 3ECTS

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

|  |  |
| --- | --- |
| Wykład: | 15h |
| Ćwiczenia: | 15h |
| Laboratorium: | 0h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Nie jest wymagana żadna wcześniejsza znajomość C ++, ale student musi mieć podstawową wiedzę o C.

**Limit liczby studentów:**

**Cel przedmiotu:**

Student powinien dobrze rozumieć takie pojęcia, jak abstrakcja, hermetyzacja, polimorfizm, klasy i interfejsy. Powinien umieć konstruować algorytmy z wykorzystaniem tych koncepcji.

**Treści kształcenia:**

Wykład:
- Powtórzenie podstaw C. Typy, rzutowanie typów, wskaźniki, tablice, wyrażenia, a także instrukcje sterujące są omawiane w ramach wspomnianych powtórzeń.
- Programowanie obiektowe - podstawy. Przedstawiono podstawowe terminy, klasy, inicjalizację obiektów, konstruktory i destruktory, jawne i niejawne ich użycie.
- Programowanie obiektowe - dostęp do treści. Omówiono specyfikatory dostępu publicznego, prywatnego i chronionego, przestrzenie nazw, rzutowanie, metody i operatory.
- Programowanie obiektowe - główne pojęcia. Przedstawiono abstrakcję, hermetyzację, dziedziczenie, polimorfizm, metody wirtualne, przeciążanie metod oraz wyjątki programowe.
Biblioteka standardowych szablonów (STL). Przedstawiono szablony, kontenery danych, atomy standardowych algorytmów i elementy numeryczne.
Ćwiczenia:
- Proste programy ilustrujące zagadnienia związane z programowaniem obiektowym, których celem jest poznanie różnych cech języka.
- Tworzenie oprogramowania przy użyciu klas i funkcji biblioteki szablonów standardowych.

**Metody oceny:**

Ocena końcowa jest obliczana jako średnia ważona wyników cząstkowych przy założeniu, że składowe oceny są pozytywne. Ocena na podstawie oceny z kolokwium teoretycznego kończącego kurs - waga: 0.5, ocena z kolokwium praktycznego kończącego ćwiczenia, waga: 0.5. Każda z dwóch składowych musi być dodatnia, tj. min. 3.0.

**Egzamin:**

nie

**Literatura:**

• P. Deitel, H. Deitel, C++ How to Program, 9th Edition, Prentice Hall, 2013
• B. Stroustrup, Programming: Principles and Practice using C++, Addison-Wesley, 2009
• B. Stroustrup, The C++ Programming Language, 4th Edition, Addison-Wesley, 2013

**Witryna www przedmiotu:**

brak

**Uwagi:**

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka CSII\_W01:**

Student posiada znajomość języka C ++

Weryfikacja:

test teoretyczny; test praktyczny

**Powiązane charakterystyki kierunkowe:** K\_W04

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o

**Charakterystyka CSII\_W02:**

Student umie wykorzystać elementy języka C ++ do wykonania prostego zadania

Weryfikacja:

test praktyczny

**Powiązane charakterystyki kierunkowe:** K\_W05

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o

**Charakterystyka CSII\_W03:**

Student umie napisać i skompilować program w języku C ++.

Weryfikacja:

test praktyczny

**Powiązane charakterystyki kierunkowe:** K\_W04

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WG.o

### Profil ogólnoakademicki - umiejętności

**Charakterystyka CSII\_U\_01:**

Student potrafi korzystać z Internetu, podręczników języka C ++ oraz informacji z innych źródeł podczas pisania programu.

Weryfikacja:

nadzór nad pracą studenta podczas ćwiczeń; egzamin praktyczny

**Powiązane charakterystyki kierunkowe:** K\_U01

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, I.P6S\_UK

**Charakterystyka CSII\_U\_02:**

Student posługuje się językiem angielskim podczas zajęć w mowie i piśmie

Weryfikacja:

Wpisz opis

**Powiązane charakterystyki kierunkowe:** K\_U04

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, I.P6S\_UK

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka CSII\_K\_01:**

Student rozumie, że na wykładach i ćwiczeniach prezentowane są tylko wybrane materiały dotyczące języka C i musi doskonalić swoje wykształcenie, aby móc sprawnie napisać program.

Weryfikacja:

rozmowa ze studentem

**Powiązane charakterystyki kierunkowe:** K\_K01

**Powiązane charakterystyki obszarowe:** I.P6S\_KO, P6U\_K, I.P6S\_KK

**Charakterystyka CSII\_K\_02:**

Student odpowiada za swoją pracę i za grupę podczas ćwiczeń, gdy próbuje rozwiązać problem i napisać program.

Weryfikacja:

nadzór nad pracą studenta podczas ćwiczeń

**Powiązane charakterystyki kierunkowe:** K\_K05

**Powiązane charakterystyki obszarowe:** I.P6S\_KO