



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Języki obiektowe - podstawy programowania [S1MiBM2>JOPP]

Przedmiot

Kierunek studiów

Mechanika i budowa maszyn

Rok/Semestr

1/1

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obieralny

Liczba godzin

Wykład

15

Laboratorium

30

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

4,00

Koordynatorzy

mgr inż. Marek Trączyński

marek.traczynski@put.poznan.pl

Wykładowcy

Wymagania wstępne

- Umiejętność obsługi komputera (praca w systemie operacyjnym, operacje na plikach i folderach). - Znajomość podstaw matematyki na poziomie szkoły średniej. - Umiejętność samodzielnego rozwiązywania problemów i myślenia algorytmicznego - Student posiada podstawowe umiejętności w zakresie programowania strukturalnego

Cel przedmiotu

Zapoznanie studentów z paradygmatem obiektowym i fundamentami programowania w językach wysokiego poziomu na przykładzie wybranego języka. W trakcie zajęć przypomniane są podstawowe pojęcia i konstrukcje występujące w programowaniu (zmienne, typy danych, instrukcje, podprogramy).

Przedmiotowe efekty uczenia się

Wiedza:

Posiada wiedzę w zakresie tworzenia i analizy algorytmów, analizy ich złożoności obliczeniowej, zapisu algorytmów w języku C++. Posiada wiedzę na temat projektowania i programowania obiektowego z użyciem narzędzi dostępnych w języku C++. Ma podstawową wiedzę o cyklach życia systemów projektowanych i implementowanych obiektowo; posiada wiedzę w zakresie obiektowych konstrukcji

systemów, posiada wiedzę o przydatnych narzędziach, bibliotekach programowania obiektowego i sposobach ich wykorzystania do budowania systemów. Zna metody i techniki rozwiązywania zadań inżynierskich poprzez odpowiednie projektowanie obiektowych modeli. Zna możliwości narzędzi programistycznych w zakresie tworzenia obiektowo-zorientowanych rozwiązań.

Umiejętności:

Ma umiejętność formułowania algorytmów i ich programowania z użyciem wybranych narzędzi programistycznych oraz potrafi ocenić złożoność obliczeniową algorytmów i problemów informatycznych. Potrafi stworzyć model obiektowy prostego systemu informatycznego; potrafi zastosować metody analizy problemu, wyodrębniania niezależnych modułów, uogólniania zagadnień dla tworzenia rozwiązań problemów w postaci obiektowo zorientowanych modeli. Potrafi dokonać identyfikacji i sformułować specyfikację prostych zadań informatyczno-inżynierskich; potrafi wykorzystywać właściwe narzędzia programistyczne w celu obsługi obiektowo zorientowanych projektów. Potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny, używając właściwych metod, technik i narzędzi. Potrafi wykonać prostą analizę sposobu funkcjonowania oprogramowania, systemu informatycznego, czy infrastruktury informatycznej.

Kompetencje społeczne:

Rozumie potrzebę uczenia się przez całe życie; potrafi inspirować i organizować proces uczenia się innych osób. Ma świadomość ważności i rozumienia pozatechnicznych aspektów i skutków działalności inżynierskiej.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wykład: Zaliczenie na podstawie testu składającego się z pytań zamkniętych wielokrotnego lub jednokrotnego wyboru oraz pytań otwartych.

Laboratorium: Zaliczenie odbywa się na podstawie sumy punktów uzyskanych za pracę na zajęciach i testów praktycznych.

Wartość oceny końcowej wynika z uzyskanego procentu sumy wszystkich punktów możliwych do zdobycia na zajęciach danego typu, zgodnie z następującą regułą:

od 51% - 3.0

od 61% - 3.5

od 71% - 4.0

od 81% - 4.5

od 91% - 5.0

Treści programowe

Podstawowe zasady i struktury programowania wysokopoziomowego. Zagadnienia związane z programowaniem obiektowym. Tworzenie programów w językach Python, C i C++. Ćwiczenie umiejętności poprzez pisanie prostych programów rozwiązujących konkretne problemy.

Tematyka zajęć

Wykład:

W pierwszej części wykładu zostaną przypomniane systemy liczbowe, typy zmiennych, tablica ASCII, podstawowe instrukcje i operatory, funkcje, architektura komputera, operacje logiczne, pojęcie algorytmu i sposoby

jego zapisu, a także podstawowe instrukcje sterujące i składnia języka. W kolejnej części omówione zostaną podstawowe pojęcia programowania obiektowego, takie jak klasa, obiekt, atrybut i metoda, a także zasady enkapsulacji, dziedziczenia i polimorfizmu. Poruszone zostaną również aspekty debugowania programów i zastosowania narzędzi AI we wspomaganie procesu programowania.

Laboratorium - praktyczne zastosowanie wiedzy poznanej na wykładzie.

Metody dydaktyczne

Wykład: prezentacja multimedialna - prezentacja ilustrowana przykładami oraz filmami, analiza problemów wraz z ich rozrysowaniem na tablicy, dyskusja i analiza problemów.

Laboratorium: Zrealizowanie zostanie poprzez pracę przy komputerach, w trakcie której wraz z prowadzącym napisane zostaną programy ilustrujące działanie omawianego zagadnienia. Na późniejszych

etapach zajęć laboratoryjnych studenci indywidualnie będą pisać indywidualnie programy zadane przez prowadzącego.

Literatura

Podstawowa:

[1] Bruce Eckel, "Thinking in C++. Edycja polska", Wydawnictwo Helion.

[2] W. Porębski, Język C++ : wprowadzenie do programowania, wyd. 2, Komputerowa Oficyna Wydawnicza "Help", Warszawa 1999.

[3] J. Grębosz, Symfonia C ++ standard : programowanie w języku C++ orientowane obiektowo, Wydawnictwo "Edition 2000" : Oficyna Kallimach, Kraków 2005.

[4] S. Prata, Język C++, wyd. 5, Wydawnictwo Helion, Gliwice 2006.

[5] A. Koenig, Accelerated C++ : practical programming by example, 22nd printing, Addison-Wesley, Boston 2013.e.

Uzupełniająca:

1. Źródła internetowe np. https://www.w3schools.com/cpp/cpp_intro.asp

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	45	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	55	2,00