



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Języki obiektowe - podstawy programowania [S1MiBM2>JOPP]

Przedmiot

Kierunek studiów

Mechanika i budowa maszyn

Studia w zakresie (specjalność)

–

Poziom studiów

pierwszego stopnia

Forma studiów

stacjonarne

Rok/Semestr

1/1

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

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.

Cel przedmiotu

Celem przedmiotu jest przedstawienie metodologii projektowania i programowania obiektowego, pewnych znanych technologii obiektowych oraz specyfiki języka C++.

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 pisemnego kolokwium składającego się z 30 pytań testowych w skali 0/1. Zaliczenie w przypadku uzyskania minimum 51 %.

Laboratorium: Zaliczenie odbywa się na podstawie kolokwium praktycznego na laboratoriach za 60 punktów oraz zadań realizowanych na laboratorium - 40 punktów. Ocena końcowa jest wyznaczana zgodnie z następującą punktacją: 100 punktów zgodnie z przyjętym systemem oceniania - 3.0 od 41 punktów, 3.5 od 56 punktów, 4.0 od 71 punktów, 4.5 od 81, 5.0 od 91 punktów

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:

Program wykładu obejmuje następujące zagadnienia: podstawy programowania obiektowego, różnice pomiędzy metodami programowania (strukturalne, obiektowe, modularne, itp.), podstawowe pojęcia i charakterystyka języka C++, pojęcie klasy, metody, definicje obiektu, obiektowe postrzeganie rzeczywistości, dziedziczenie klas, przeciążanie klas i metod, zaprzyjaźnienie klas i metod, wzorce obiektowe, klasy abstrakcyjne, metody wirtualne, przeciążanie operatorów, strumienie, wątki, wyjątki, Podstawowe pojęcia i charakterystyka języka Java, aplikacje, aplety, programowanie w środowisku tekstowym i graficznym, biblioteki związane z programowaniem kontekstu graficznego

.Laboratorium - praktyczne zastosowanie wiedzy poznanej na wykładzie, implementacja i testowanie - omówienie tematów:

1. Projektowanie obiektowe - kapsułkowanie, ochrona danych
2. Klasy i obiekty, ochrona danych przez modyfikatory dostępu
3. Tworzenie i destrukcja obiektów i tablic obiektów
4. Polimorfizm - przeciążanie operatorów
5. Dziedziczenie: a. Klasy bazowe i pochodne, hierarchie klas; b. Dziedziczenie wielokrotne, dziedziczenie wirtualne
6. Abstrakcyjne typy danych: a. Metody wirtualne, klasy abstrakcyjne, wirtualne destruktory; b. Interfejsy a klasy abstrakcyjne
7. Wzorce (ang. templates) funkcji i klas
8. Obsługa wyjątków
9. Specyfika C++: a. l-wartości i r-wartości, efektywne przekazywanie argumentów itp.; b. Przestrzenie nazw; c. RTTI - runtime type identification
10. Rozszerzenia języka C++ w standardach C++11, C++14, C++17 i C++20

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 pisać będą programy realizujące dane zadanie, które wcześniej będzie omówione i przykładowo zaprezentowane 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