

Programowanie obiektowe - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Programowanie obiektowe
Kod przedmiotu	11.3-WF-FizP-ProOb-L-S14_genNLL9Z
Wydział	Wydział Fizyki i Astronomii
Kierunek	Fizyka
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. licencjata
Semestr rozpoczęcia	semestr zimowy 2017/2018

Informacje o przedmiocie	
Semestr	3
Liczba punktów ECTS do zdobycia	6
Typ przedmiotu	obowiązkowy
Język nauczania	polski
Sylabus opracował	• dr Marcin Kośmider

Formy zajęć					
Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Wykład	30	2	-	-	Egzamin
Laboratorium	30	2	-	-	Zaliczenie na ocenę

Cel przedmiotu

Przedmiot stanowi wprowadzenie do programowania obiektowego i zastosowania technik obiektowych w rozwiązywaniu problemów i zagadnień z fizyki i nauk pokrewnych. Celem przedmiotu jest przedstawienie technik programowania obiektowego w języku C++ oraz przedstawienie programowania obiektowego jako sposobu analizy i modelowania problemu w oderwaniu od specyfiki danego języka programowania. Kurs oprócz omówienie technik obiektowych zakłada również zapoznanie uczestników z standardami kodowania, formatowania kodu oraz jego dokumentowania i utrzymania w celu jego rozwoju.

Wymagania wstępne

Minimalny zakres wymagań wstępnych to sprawne posługiwanie się systemem Linux (zarówno w konsoli jak i w środowisku graficznym), znajomość podstaw programowania obejmujących programowanie proceduralne.

Zakres tematyczny

1. Wstęp do programowania obiektowego
 - programowanie obiektowe a programowanie strukturalne
 - pojęcie klasy, obiektu, metod i pól
 - pojęcie konstruktora i destruktora
 - instancje klas
 - enkapsulacja danych
 - wskaźniki do obiektów klas
 - przeładowanie operatorów
 - funkcje zaprzyjaźnione
2. Praca z predefiniowanymi klasami
 - przypomnienie operacji IO
 - omówienie wybranych kontenerów z biblioteki STL
3. Wskaźniki
 - obiekty i dynamiczna rezerwacja pamięci
 - konstruktor kopiujący
 - destruktor„inteligentne” wskaźniki
4. Dziedziczenie, polimorfizm i „code reuse”
 - dziedziczenie
 - metody wirtualne i klasy abstrakcyjne
 - pojęcie interfejsu
 - polimorfizm
 - koncepcja „code reuse”

5. Organizacja kodu
 - standardy nazewnictwa i formatowania kodu
 - pliki nagłówkowe
 - przestrzenie nazw
 - makefile
 - dokumentowanie kodu
 - systemy kontroli wersji
6. Szablony
7. Wyjątki
8. Modelowanie i programowanie obiektowe
 - analiza problemu i budowa jego modelu
 - proces powstawania oprogramowania
 - diagramy UML
 - kodowanie diagramów UML w C++
9. Wzorce projektowe
 - koncepcja wzorców projektowych
 - wzorce kreacyjne
 - wzorce strukturalne
 - wzorce czynnościowe
10. Frameworki
 - pojęcie frameworku i zastosowania
 - przykład frameworku - Qt

Metody kształcenia

Wykład:

Wykład konwencjonalny, problemowy, pokaz, dyskusja, warsztaty (testowanie aktualnie omawianych fragmentów kodu), burza mózgów

Laboratoria:

Ćwiczenia laboratoryjne, metoda projektu, praca w grupie, giełda pomysłów, burza mózgów, prezentacja, praca z dokumentacją, samodzielne pozyskiwanie wiedzy

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Student potrafi skompilować i uruchomić program oraz dokonać analizy i interpretacji uzyskanych wyników dotyczących rozpatrywanego problemu z fizyki, jak również zweryfikować poprawność jego działania na podstawie posiadanej wiedzy fizycznej o rozważanym problemie.	K1A_W04 K1A_U04 K1A_U07	- bieżąca kontrola na zajęciach - dyskusja - egzamin - ustny, opisowy, testowy i inne - projekt	- Wykład - Laboratorium
Student potrafi samodzielnie wyszukiwać i stosować narzędzia i informacje pomocne do rozwiązania postawionego problemu.	K1A_W09 K1A_U07 K1A_K01 K1A_K04	- bieżąca kontrola na zajęciach - dyskusja - egzamin - ustny, opisowy, testowy i inne - projekt	- Wykład - Laboratorium
Student potrafi zdefiniować i objaśnić postawiony problem dokonując jego rozbicia na problemy elementarne, potrafi dokonać obiektowej analizy problemu, wskazać modele, obiekty i zachodzące relacje między nimi.	K1A_W03 K1A_U05 K1A_U07	- bieżąca kontrola na zajęciach - dyskusja - egzamin - ustny, opisowy, testowy i inne - projekt	- Wykład - Laboratorium
Student potrafi zastosować posiadaną wiedzę z zakresu programowania i modelowania obiektowego oraz dostępne narzędzia do przedstawienia sposobów rozwiązania problemu z fizyki i dziedzin pokrewnych w postaci kodu źródłowego programu.	K1A_W04 K1A_W09 K1A_U04 K1A_U05	- bieżąca kontrola na zajęciach - dyskusja - egzamin - ustny, opisowy, testowy i inne - projekt	- Wykład - Laboratorium

Opis efektu	Symbolo efektów	Metody weryfikacji	Forma zajęć
Student potrafi stworzyć i zaprezentować sprawozdanie z powierzonego projektu.	• K1A_U08 • K1A_K01 • K1A_K04	• bieżąca kontrola na zajęciach • dyskusja • egzamin - ustny, opisowy, testowy i inne • projekt	• Laboratorium
Student zna regulamin i zasady BHP obowiązujące w pracowni komputerowej.	• K1A_W06	• bieżąca kontrola na zajęciach • dyskusja • projekt	• Laboratorium
Student potrafi współpracować w grupie, czuje się odpowiedzialny za powierzone mu zadania, jest otwarty na nowe koncepcje i pomysły.	• K1A_K01 • K1A_K02	• bieżąca kontrola na zajęciach • dyskusja • projekt	• Wykład • Laboratorium

Warunki zaliczenia

Wykład:

Egzamin praktyczny polegający na rozwiązaniu postawionego problemu (wylosowanego z listy problemów). Ocenie końcowej podlega analiza problemu, przedstawienie algorytmów rozwiązania problemu, kod źródłowy oraz ocena i weryfikacja uzyskanych wyników

Laboratoria

Ocena końcowa składa się z: średniej ocen uzyskanych w trakcie laboratoriów z aktywności i krótkich testów sprawdzających postępy w nauce (50% oceny końcowej), oceny projektu semestralnego (50 % oceny końcowej). Warunkiem zaliczenia projektu semestralnego jest jego wykonanie, przygotowanie i oddanie w przewidzianym terminie sprawozdania z projektu oraz jego prezentacja. Przed przystąpieniem do egzaminu student musi uzyskać zaliczenie z ćwiczeń. Ocena końcowa: średnia ważona ocen z egzaminu (60%) i ćwiczeń (40%).

Literatura podstawowa

- [1] Bruce Eckel, Thinking in C++ Edycja Polska, Helion Gliwice, 2002.
 [2] Bruce Eckel, Thinking in C++ Edycja Polska, Tom 2, Helion Gliwice, 2004.
 [3] Steve Holzner, Design patterns for dummies, Willey Publishing Ing. Indianapolis 2006.

Literatura uzupełniająca

- [1] Internet

Uwagi

Wykład powinien odbywać się w sali z dostępem do internetu. Laboratoria komputerowe powinny odbywać się w grupach umożliwiających samodzielną pracę przy komputerze każdego studenta i nie liczniejszych niż 12 osób.

Zmodyfikowane przez prof. dr hab. Mirosław Dudek (ostatnia modyfikacja: 13-09-2017 11:08)

Wygenerowano automatycznie z systemu SyllabUZ