Object oriented programming - opis przedmiotu

Informacje ogólne

Nazwa przedmiotu	Object oriented programming
Kod przedmiotu	13.2-WF-FizP-OP-S17
Wydział	~~Wydział Fizyki i Astronomii~~
Kierunek	Fizyka
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. licencjata
Semestr rozpoczęcia	semestr zimowy 2018/2019

Informacje o przedmiocie

Semestr	3
Liczba punktów ECTS do zdobycia	6
Typ przedmiotu	obowiązkowy
Język nauczania	angielski
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	15	1	-	-	Egzamin
Laboratorium	45	3	-	-	Zaliczenie na ocenę

Cel przedmiotu

The aim of this course is to introduce the Object Oriented Programming techniques required to develop and create modern applications related to the „every day” and science problems. This is an active course where students solve realistic problems from beginning. Students learn how to analyse problem in the object oriented way and how to implement code according to the standards.

Wymagania wstępne

The efficient use of the Linux system (both in the terminal and in the graphical environment), knowledge of the basics of programming including procedural programming.

Zakres tematyczny

1. Introduction

- object and procedural programming

- class, object and methods

- constructor and destructor

- encapsulation

- pointers

- operators overloading

- friend function

2. Using standard class

- IO operations

- short introduction to the STL containers and algorithms

3. Pointers

- objects and dynamic memory allocation

- copy constructor

- destructor

- „intelligent” pointers

4. Inheritance, polymorphism and code reuse

- inheritance

- virtual and abstract classes and methods

- interfaces

- polymorphism

- the idea of „code reuse”

5. Clean code

- name standards

- header files

- namespaces

- makefile

- code comment and documentation

- version control systems

6. Templates in C++

7. Exception

8. Object oriented modelling and programming

- defining and analysing problem and model creation

- UML diagrams

- coding UML diagrams in C++

9. Design patterns

- the idea

- creational patterns

- structural patterns

- behaviour patterns

10. Frameworks

- the idea

- Qt as as sample

Metody kształcenia

Lecture:

Convencional lecture, work with problems, discusiion, workshop

Laboratory:

Laboratory exercise, project, work in group, presentation, work with documentation, independed work, brain storm

Efekty uczenia się i metody weryfikacji osiągania efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Student knows how to search, find and use modern tools and informations that can be used to solve given problem		• bieżąca kontrola na zajęciach • dyskusja • egzamin - ustny, opisowy, testowy i inne • projekt	• Wykład • Laboratorium

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
The student can apply his knowledge of programming and object modeling and available tools to present ways to solve the considered problem in physics or related fields in the form of the source code of the program.		• bieżąca kontrola na zajęciach • dyskusja • egzamin - ustny, opisowy, testowy i inne • projekt	• Wykład • Laboratorium
Student can create and present report from given problem		• bieżąca kontrola na zajęciach • dyskusja • egzamin - ustny, opisowy, testowy i inne • projekt	• Laboratorium
The student is able to cooperate in a group, feels responsible for the tasks entrusted to him, is open to new concepts and ideas.		• bieżąca kontrola na zajęciach • dyskusja • projekt	• Wykład • Laboratorium
Student know laboratory statute and BHP rules		• bieżąca kontrola na zajęciach • dyskusja • projekt	• Laboratorium
Student can compile and run program. For a given physical problem student can analyse and interpret computational resuts and verify the correctness of a written application		• bieżąca kontrola na zajęciach • dyskusja • egzamin - ustny, opisowy, testowy i inne • projekt	• Wykład • Laboratorium
Student can define problem and explain the problem posed by breaking it into elementary problems, describes and analyses it in the object oriented way, indicating the models, objects and relations between them.		• bieżąca kontrola na zajęciach • dyskusja • egzamin - ustny, opisowy, testowy i inne • projekt	• Wykład • Laboratorium

Warunki zaliczenia

Lecture:
A practical exam consisting in solving a given problem (chosen from the list of problems). Final evaluation is subject to problem analysis, presentation of problem solving algorithms, source code as well as evaluation and verification of obtained results

Laboratory:
The final grade consists of: average marks obtained during laboratories with activity and short tests to check learning progress (50% of final grade), semester project assessment (50% of final grade). The condition for passing the semester project is its implementation, preparation and delivery of the project report and its presentation within the prescribed period. Before taking the exam the student must get a pass from the exercises.

Final grade: weighted average of exam grades (60%) and exercises (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

The lecture should take place in a room with Internet access. Computer laboratories should take place in groups enabling independent work at the computer of every student and not more than 12 people.

Zmodyfikowane przez dr hab. Piotr Lubiński, prof. UZ (ostatnia modyfikacja: 01-08-2018 14:45)