Object-oriented programming - opis przedmiotu

Informacje ogólne

Nazwa przedmiotu	Object-oriented programming
Kod przedmiotu	11.3-WE-INFP-OOP-Er
Wydział	Wydział Nauk Inżynieryjno-Technicznych
Kierunek	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	Program Erasmus pierwszego stopnia
Semestr rozpoczęcia	semestr zimowy 2021/2022

Informacje o przedmiocie

Semestr	2
Liczba punktów ECTS do zdobycia	5
Typ przedmiotu	obowiązkowy
Język nauczania	angielski
Sylabus opracował	• dr hab. inż. Paweł Majdzik, prof. UZ

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

To provide basic knowledge about object programming paradigms.

To provide basic knowledge about abstract data typing definition with member functions (encapsulation),

To provide basic knowledge about inheritance, polymorphism and virtual functions, templates of classes and functions.

To give basic skills in designing programs and utilizing tools (e.g. tools from Standard Template Library).

Wymagania wstępne

Principles of programming, Algorithms and data structures

Zakres tematyczny

Introduction to object programming. Concept of abstract data typing. Class definition. Encapsulation – declaration and definition of class member methods. Private and public class members. Constructors and destructors. Default and copy constructors. Synthesized constructors. Destructors.

Operators overloading. User defined conversions: converting function, converting constructor. Functions overloading: friend functions and inline functions, constructor and operator conversion.

Inheritance rules. Inheritance and the composition of objects. Protected members. Multiple and multi-base inheritance. Problem of variable names in multi-base inheritance. Polymorphism. Polymorphism. Virtual functions. Pure virtual functions. Early and late binding. Time and memory costs connected with application of polymorphism. Abstract classes - defining and examples of abstract classes application in object-oriented programs.

Standard Template Library. Function templates. Specialized functions. Phases of function adjustment. Class templates. Definition of class templates. Class templates versus microdefinitions. Containers and algorithms, iterators, associative containers, function objects. Designing of object-oriented programming. Design pattern .

Adapter pattern, facade pattern, bridge pattern etc..

Metody kształcenia

Lectures, laboratory exercises.

Efekty uczenia się i metody weryfikacji osiągania efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Knows basic design templates and understands their meanings in flexible software design.		• aktywność w trakcie zajęć • bieżąca kontrola na zajęciach • egzamin - ustny, opisowy, testowy i inne	• Wykład • Laboratorium
Can define and implement basic integral class elements: constructors, operator functions, destructors		• bieżąca kontrola na zajęciach • dokumentacja praktyki	• Wykład • Laboratorium

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Student is able to design and implement simple object programs		• aktywność w trakcie zajęć • konspekt	• Laboratorium
Understands basic concepts related to object programming: encapsulation, homogeneity		• bieżąca kontrola na zajęciach • egzamin - ustny, opisowy, testowy i inne	• Wykład • Laboratorium
Can define and implement basic integral class elements: constructors, operator functions, destructors		• bieżąca kontrola na zajęciach • egzamin - ustny, opisowy, testowy i inne	• Wykład • Laboratorium

Warunki zaliczenia

Lecture – the passing condition is to obtain a positive mark from the examination.

Laboratory – the passing condition is to obtain positive marks from all laboratory exercises to be planned during the semester.

Calculation of the final grade: lecture 50% + laboratory 50%

Literatura podstawowa

Eckel B.: Thinking in C++, Prentice Hall, US Ed edition, 2002 2.

Stroustrup B.: The C++ Programming Language, Addison – Wesley, 2004

Literatura uzupełniająca

Lippman S.B.: Inside the C++ Object Model, Addison – Wesley, 1996

Uwagi

Zmodyfikowane przez prof. dr hab. inż. Andrzej Obuchowicz (ostatnia modyfikacja: 13-09-2021 11:39)