

OBJECT-ORIENTED PROGRAMMING - course description

General information	
Course name	OBJECT-ORIENTED PROGRAMMING
Course ID	11.3-WF-FizP-ProOb-L-S14_genNLL9Z
Faculty	Faculty of Physics and Astronomy
Field of study	Physics
Education profile	academic
Level of studies	First-cycle studies leading to Bachelor's degree
Beginning semester	winter term 2021/2022

Course information	
Semester	3
ECTS credits to win	6
Available in specialities	Computer Physics
Course type	obligatory
Teaching language	polish
Author of syllabus	dr Marcin Kośmider

Classes forms					
The class form	Hours per semester (full-time)	Hours per week (full-time)	Hours per semester (part-time)	Hours per week (part-time)	Form of assignment
Lecture	15	1	-	-	Exam
Laboratory	45	3	-	-	Credit with grade

Aim of the course

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 Python 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.

Prerequisites

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

Scope

1. Wstęp do programowania obiektowego
 - programowanie obiektowe, a programowanie strukturalne
 - pojęcie klasy, obiektu, metod i pól
 - pojęcie konstruktora i destruktora
 - instancje klas
 - enkapsulacja danych
 - funkcje specjalne
 - metody i atrybuty klasy, metody statyczne
2. Dziedziczenie, polimorfizm
 - dziedziczenie
 - polimorfizm
 - abstrakcja
3. Organizacja kodu
 - standardy nazewnictwa i formatowania kodu
 - moduły i pakiety
 - przestrzenie nazw
 - dokumentowanie kodu
 - systemy kontroli wersji
4. Modelowanie i programowanie obiektowe
 - analiza problemu i budowa jego modelu
 - proces powstawania oprogramowania
 - diagramy UML

5. Wzorce projektowe

- koncepcja wzorców projektowych
- wzorce kreacyjne
- wzorce strukturalne
- wzorce czynnościowe

6. Frameworki

- pojęcie frameworku i zastosowania
- wybrany przykład frameworku

Teaching methods

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

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
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	• a discussion • a project • an exam - oral, descriptive, test and other • an ongoing monitoring during classes	• Lecture • Laboratory
Student potrafi samodzielnie wyszukiwać i stosować narzędzia i informacje pomocne do rozwiązania postawionego problemu.	• K1A_W09 • K1A_U07 • K1A_K01 • K1A_K04	• a discussion • a project • an exam - oral, descriptive, test and other • an ongoing monitoring during classes	• Lecture • Laboratory
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	• a discussion • a project • an exam - oral, descriptive, test and other • an ongoing monitoring during classes	• Lecture • Laboratory
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	• a discussion • a project • an exam - oral, descriptive, test and other • an ongoing monitoring during classes	• Lecture • Laboratory
Student potrafi stworzyć i zaprezentować sprawozdanie z powierzonego projektu.	• K1A_U08 • K1A_K01 • K1A_K04	• a discussion • a project • an exam - oral, descriptive, test and other • an ongoing monitoring during classes	• Laboratory
Student zna regulamin i zasady BHP obowiązujące w pracowni komputerowej.	• K1A_W06	• a discussion • a project • an ongoing monitoring during classes	• Laboratory

Outcome description	Outcome symbols	Methods of verification	The class form
Student potrafi współpracować w grupie, czuje się odpowiedzialny za powierzone mu zadania, jest otwarty na nowe koncepcje i pomysły.	• K1A_K01 • K1A_K02	• a discussion • a project • an ongoing monitoring during classes	• Lecture • Laboratory

Assignment conditions

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%).

Recommended reading

1. "Zaawansowany Python. Jasne, zwięzłe i efektywne programowanie", R.Luciano, APN Promise
2. "Learning Python. Powerful Object-Oriented Programming. 5th Edition", M.Lutz, Helion
3. "Python 3 Object-Oriented Programming" D.Philips, Helion
4. "Clean Code in Python" M.Anaya, Helio
5. Steve Holzner, Design patterns for dummies, Willey Publishing Ing. Indianapolis 2006.

Further reading

[1] Internet

Notes

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 licznieszych niż 12 osób.

Modified by dr Marcin Kośmider (last modification: 10-05-2021 21:12)

Generated automatically from SylabUZ computer system