

Podstawy programowania - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Podstawy programowania
Kod przedmiotu	11.3-WF-FizP-PPr-S16
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	2
Liczba punktów ECTS do zdobycia	5
Typ przedmiotu	obowiązkowy
Język nauczania	polski
Sylabus opracował	prof. dr hab. Mirosław Dudek

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	60	4	-	-	Zaliczenie na ocenę

Cel przedmiotu

Celem przedmiotu jest nauka podstaw programowania oraz umiejętność wykorzystania zdobytej wiedzy do rozwiązywania różnorodnych problemów z szczególnym uwzględnieniem problemów dotyczących nauk ścisłych. Takie podejście do programowania wymaga zrozumienia nie tylko składni języka programowania, ale również podstaw algorytmiki, faz tworzenia oprogramowania, standardów kodowania, umiejętności pracy z dokumentacją oraz analizy i redukcji złożonego problemu do szeregu problemów elementarnych. Podstawy programowania stanowią również niezbędny fundament dla zrozumienia kolejnych przedmiotów komputerowych takich jak m.in. metody numeryczne, programowanie obiektowe czy modelowanie i symulacje komputerowe.

Wymagania wstępne

Zakłada się, że uczestnicy zajęć posiadają podstawową wiedzę z zakresu pracy w środowisku Linux (umiejętność pracy w terminalu, podstawowa znajomość dowolnie wybranego edytora tekstu z opcją kolorowania składni). Wymagania wstępne stanowią podzbiór materiału realizowanego na pierwszej pracowni komputerowej.

Zakres tematyczny

1. Zasady bezpieczeństwa i higieny pracy, regulamin pracowni komputerowej

2. Wstęp do programowania:

- pojęcie algorytmu, strategia „dziel i zwyciężaj”
- cykl „analiza – kod – wykonanie”
- rodzaje błędów
- kod źródłowy, maszynowy, kompilacja, linkowanie
- historia języków programowania

3. Wstęp do języka C

- typy zmienny, błędy nadmiaru i niedomiaru,
- deklaracje i definicje zmiennych, rzutowanie typów
- zmienne lokalne i globalne
- wyprowadzanie danych na ekran
- operatory matematyczne
- składnia języka C
- konwencje nazewnictwa i formatowania kodu źródłowego
- kompilowanie i linkowanie

- 4. Warunki
 - operatory logiczne
 - operator warunkowy
 - instrukcja warunkowa
- 5. Pętle
 - pętla while
 - pętla do while
 - pętla for
- 6. Tablice
 - tablice jednowymiarowe
 - tablice wielowymiarowe
- 7. Funkcje
 - pojęcie funkcji
 - parametry przekazywane do funkcji, parametry domyślne, wartości zwracane
 - zakres ważności zmiennych
 - referencje
 - rekurencja
 - praca z tablicami
- 8. Wskaźniki i zmienne cz.2.
 - wskaźniki
 - referencje
 - przedrostki static i const
 - dynamiczna alokacja pamięci
 - tablice wskaźników
 - parametry przekazywane w trakcie uruchomienia programu
- 9. Struktury
 - pojęcie struktury
 - struktury jako typy danych
 - wykorzystanie struktur
- 10. Operacje IO
 - binarne i tekstowe strumienie danych
 - standardowe strumienie wejścia i wyjścia
 - czytanie danych z plików
 - zapis do plików

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 ma świadomość istnienia oprogramowania Open Source stanowiącego profesjonalną alternatywę dla oprogramowania komercyjnego. Student ma świadomość szybkości zmian w branży IT i związaną z tym konieczność ustawicznego podnoszenia swoich kompetencji. Student potrafi stworzyć i zaprezentować sprawozdanie z powierzonego projektu.	• K1A_W09 • K1A_U08 • K1A_K01 • K1A_K04 • K1A_K06	• dyskusja	• Laboratorium
Student potrafi samodzielnie wyszukiwać i stosować narzędzia i informacje pomocne do rozwiązania postawionego problemu.	• K1A_W09 • K1A_U07	• sprawdzian	• Laboratorium
Student potrafi zdefiniować i objaśnić postawiony problem dokonując jego rozbicia na problemy elementarne oraz przedstawić metody (algorytmy) optymalnego rozwiązania problemu.	• K1A_W03 • K1A_U05 • K1A_U07	• egzamin - ustny, opisowy, testowy i inne • sprawdzian	• Wykład • Laboratorium
Student potrafi współpracować w grupie, czuje się odpowiedzialny za powierzone mu zadania, jest otwarty na nowe koncepcje i pomysły.	• K1A_K02 • K1A_K03	• dyskusja	• Laboratorium

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Student zna typy danych, instrukcje sterujące, funkcje, potrafi pracować z statycznymi i dynamicznymi tablicami oraz strumieniami IO. Potrafi zastosować posiadaną wiedzę i dostępne narzędzia do przedstawienia rozwiązania problemu (w szczególności z zakresu fizyki i dziedzin pokrewnych) w postaci kodu źródłowego	• K1A_W04 • K1A_W09 • K1A_U04 • K1A_U05	• egzamin - ustny, opisowy, testowy i inne • sprawdzian	• Wykład • Laboratorium
Student potrafi skompilować i uruchomić program oraz dokonać analizy i interpretacji uzyskanych wyników dotyczących problemów z dziedziny fizyki (lub pokrewnych), jak również zweryfikować poprawność jego działania na podstawie posiadanej wiedzy z danej dziedziny.	• K1A_W04 • K1A_U04 • K1A_U07	• sprawdzian	• Laboratorium
Student zna regulamin i zasady BHP obowiązujące w pracowni komputerowej.	• K1A_W06	• sprawdzian	• 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: średnia ocen uzyskanych w trakcie laboratoriów z aktywności i krótkich testów sprawdzających postępy w nauce (50% oceny końcowej) oraz 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 laboratoryjnych.

Ocena końcowa: średnia ważona ocen z egzaminu (60%) i ćwiczeń (40%).

Literatura podstawowa

[1] Allen Downey, Think Python. How to Think Like a Computer Scientist, 2013, Green Tea Press Needham, Massachusetts.

[2] Stephen Prata „Język C, Szkoła programowania”, Helion 2016

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 dr hab. Piotr Lubiński, prof. UZ (ostatnia modyfikacja: 27-06-2018 16:55)

Wygenerowano automatycznie z systemu SylabUZ