

Fundamentals of programming - course description

General information	
Course name	Fundamentals of programming
Course ID	11.3-WF-FizP-PPr-S16
Faculty	Faculty of Physics and Astronomy
Field of study	Astronomy
Education profile	academic
Level of studies	First-cycle studies leading to Bachelor's degree
Beginning semester	winter term 2018/2019

Course information	
Semester	2
ECTS credits to win	5
Course type	obligatory
Teaching language	polish
Author of syllabus	prof. dr hab. Mirosław Dudek

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	-	-	Credit with grade
Laboratory	60	4	-	-	Credit with grade

Aim of the course

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.

Prerequisites

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.

Scope

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 zmiennych, 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

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 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.	• K_W09 • K_K01 • K_K02 • K_K06	• a discussion	• Laboratory
Student potrafi samodzielnie wyszukiwać i stosować narzędzia i informacje pomocne do rozwiązania postawionego problemu.	• K_W09 • K_U10	• a quiz	• Laboratory
Student potrafi zdefiniować i objaśnić postawiony problem dokonując jego rozbicia na problemy elementarne oraz przedstawić metody (algorytmy) optymalnego rozwiązania problemu.	• K_W03 • K_W05 • K_U05 • K_U10	• a quiz • an exam - oral, descriptive, test and other	• Lecture • Laboratory
Student potrafi współpracować w grupie, czuje się odpowiedzialny za powierzone mu zadania, jest otwarty na nowe koncepcje i pomysły.	• K_K03 • K_K04	• a discussion	• Laboratory
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 dziedziny pokrewnych) w postaci kodu źródłowego	• K_W08 • K_W09 • K_U05 • K_U06	• a quiz • an exam - oral, descriptive, test and other	• Lecture • Laboratory

Outcome description	Outcome symbols	Methods of verification	The class form
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.	• K_W08 • K_U07 • K_U10	• a quiz	• Laboratory
Student zna regulamin i zasady BHP obowiązujące w pracowni komputerowej.	• K_W11	• a quiz	• 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: 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%).

Recommended reading

[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

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 licniejszych niż 12 osób.

Modified by dr Magdalena Szkudlarek (last modification: 10-05-2018 13:39)

Generated automatically from SylabUZ computer system