

Podstawy programowania - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Podstawy programowania
Kod przedmiotu	06.4-WI-GeoTSP-PP-S18
Wydział	Wydział Budownictwa, Architektury i Inżynierii Środowiska
Kierunek	Geoinformatyka i techniki satelitarne
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. inżyniera
Semestr rozpoczęcia	semestr zimowy 2021/2022

Informacje o przedmiocie	
Semestr	1
Liczba punktów ECTS do zdobycia	5
Typ przedmiotu	obowiązkowy
Język nauczania	polski
Sylabus opracował	dr inż. Arkadiusz Denisiewicz

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	9	0,6	Egzamin
Laboratorium	45	3	27	1,8	Zaliczenie na ocenę

Cel przedmiotu

Celem przedmiotu jest nauka podstaw programowania strukturalnego (na przykładzie języków Fortran, C/C++) i elementów programowania obiektowego z wykorzystaniem gotowych klas w języku C/C++ oraz umiejętność wykorzystania zdobytej wiedzy do rozwiązywania różnorodnych problemów z szczególnym uwzględnieniem problemów dotyczących nauk ścisłych i technicznych. 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 pojawiających się na kolejnych etapach nauczania.

Wymagania wstępne

Umiejętność obsługi komputera na poziomie pozwalającym na swobodne posługiwanie się zainstalowanym oprogramowaniem i poruszaniem się w strukturze katalogów. Średnia biegłość w pisaniu na klawiaturze.

Zakres tematyczny

1. Zasady BHP i regulamin pracowni komputerowej

2. Wstęp do programowania

- pojęcie algorytmu, strategia dziel i zwyciężaj
- cykl życia programu
- rodzaje błędów
- kod źródłowy, maszynowy, kompilacja, linkowanie
- historia języków i technik programowania

3. Wstęp do języków C i Fortran

- typy zmiennych, reprezentacja liczb, błędy nadmiaru i niedomiaru
- deklaracje i definicje zmiennych, rzutowanie typów, błędy
- zmienne lokalne i globalne
- wypisywanie danych na ekranie (printf, write)
- operatory matematyczne
- składnia języków C, Fortran i standardy kodowania
- kompilowanie i linkowanie

4. Warunki

- operatory logiczne
- operator warunkowy
- instrukcja if .. else
- budowanie wyrażeń logicznych

5. Pętle

- pętla while
- pętla do .. while

- pętla for

6. Tablice statyczne

- pojęcie stałej i konwencja nazewnictwa
- tablice jednowymiarowe
- przykłady algorytmów wykorzystujących tablice jednowymiarowe
- c-string jako reprezentacja tablicy znaków
- biblioteka cstring
- referencje i kopiowanie tablic
- pobieranie danych od użytkownika – scanf, read
- tablice wielowymiarowe

7. Funkcje

- pojęcie i funkcji i jej zastosowanie
- deklaracja i definicja funkcji
- zakres ważności zmiennych,
- zmienne typu const i static
- parametry funkcji – referencje, wartości, parametry domyślne
- tablice jednowymiarowe jako parametry wywołania funkcji
- tablice dwuwymiarowe jako parametry wywołania funkcji, dyrektywa #DEFINE
- rekurencje
- przeglądowanie nazw funkcji

8. Struktury

- struktury jako typy danych
- wykorzystanie struktur w pracy z funkcjami

9. Pliki nagłówkowe i źródłowe

- organizacja kodu
- preprocesor i jego dyrektywy
- pliki nagłówkowe i źródłowe
- kompilacja

9. Dynamiczna alokacja pamięci (C++)

- wskaźniki i referencje
- dynamiczna alokacja pamięci i jej zwalnianie
- dynamiczne tablice jednowymiarowe
- wskaźniki jako parametry wywołania funkcji
- zwracanie wskaźników z funkcji
- dynamiczna alokacja pamięci w funkcjach
- problemy wycieku pamięci
- dynamiczne tablice wielowymiarowe

10. Obiektowe operacje IO (C++)

- pojęcie klasy i obiektu, metody , pola
- obiekt cin i cou- formatowane i nieformatowane operacje na przykładzie cin, cout
- pojęcie strumienia danych jako warstwy abstrakcji
- tekstowe operacje plikowe jako operacje strumieniowe
- błędy operacji IO

Metody kształcenia

Metody podające: wykład informacyjny z wykorzystaniem technik multimedialnych, wykład problemowy.

Metody poszukujące: laboratoria ćwiczeniowo-praktyczne, projekt, dyskusja, praca w grupach

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

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 geoinformatyki i dziedzin pokrewnych) w postaci kodu źródłowego	• K_W04 • K_W07 • K_W09 • K_U02 • K_U04 • K_U06 • K_K01 • K_K03	• aktywność w trakcie zajęć • bieżąca kontrola na zajęciach • egzamin - ustny, opisowy, testowy i inne • obserwacja i ocena aktywności na zajęciach • obserwacje i ocena umiejętności praktycznych studenta	• Wykład • Laboratorium

Opis efektu	Symbol e efektów	Metody weryfikacji	Forma zajęć
Student potrafi skompilować i uruchomić program oraz dokonać analizy i interpretacji uzyskanych wyników dotyczących problemów z dziedziny geoinformatyki (lub pokrewnych), jak również zweryfikować poprawność jego działania na podstawie posiadanej wiedzy z danej dziedziny.	K_W01 K_W02 K_W03 K_W04 K_W07 K_U01 K_U02 K_U05 K_U06 K_K01 K_K02 K_K03	- egzamin - ustny, opisowy, testowy i inne - obserwacja i ocena aktywności na zajęciach - obserwacje i ocena umiejętności praktycznych studenta	- Wykład - Laboratorium
Student potrafi samodzielnie wyszukiwać i stosować narzędzia i informacje pomocne do rozwiązania postawionego problemu.	K_W04 K_W07 K_U01 K_U02 K_U04 K_U06 K_K01 K_K04	- obserwacja i ocena aktywności na zajęciach - obserwacje i ocena umiejętności praktycznych studenta	Laboratorium
Student potrafi zdefiniować i objaśnić postawiony problem dokonując jego rozbicia na problemy elementarne oraz przedstawić metody (algorytmy) optymalnego rozwiązania problemu.	K_W01 K_W02 K_W03 K_W07 K_U01 K_U04 K_U05 K_U06 K_K01 K_K03	- aktywność w trakcie zajęć - bieżąca kontrola na zajęciach - egzamin - ustny, opisowy, testowy i inne - obserwacje i ocena umiejętności praktycznych studenta	- Wykład - 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

- Język C. Szkoła programowania. Wydanie VI, Stephen Prata, Helion 2016
- Podstawy języka C++, Stanley B. Lippman, WNT 2003
- Programowanie w języku Fortran 90 i 95, J.R. Piechna, OWPW 2000.

Literatura uzupełniająca

Internet

Uwagi

W trakcie prowadzenia zajęć szczególny nacisk należy położyć na elementy praktycznego programowania starając się zwrócić uwagę na poprawne formatowanie kodu, na stosowane nazewnictwo, umiejętność organizacji kodu.

Zmodyfikowane przez dr inż. Arkadiusz Denisiewicz (ostatnia modyfikacja: 24-04-2021 10:50)

Wygenerowano automatycznie z systemu SylabUZ