

Programming basics - course description

General information	
Course name	Programming basics
Course ID	06.4-WI-GeoTSP-PP-S18
Faculty	Faculty of Civil Engineering, Architecture and Environmental Engineering
Field of study	Geoinformatics and satellite technology
Education profile	academic
Level of studies	First-cycle studies leading to Engineer's degree
Beginning semester	winter term 2020/2021

Course information	
Semester	1
ECTS credits to win	5
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	9	0,6	Exam
Laboratory	45	3	27	1,8	Credit with grade

Aim of the course

Celem przedmiotu jest nauka podstaw programowania strukturalnego 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.

Prerequisites

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

Scope

1. Zasady BHP i regulamin pracowni komputerowej

2. Wstęp do programowania

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

3. Wstęp do języka C

- typy zmiennych, reprezentacja liczb, błędy nadmiaru i niedomiaru
- deklaracje i definicje zmiennych, rzutowanie typów, błędy
- zmienne lokalne i globalne
- wypisywanie danych na ekranie (printf)
- operatory matematyczne
- składnia języka C 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
- 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
- przeładowanie 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 cout- formatowane i nieformatowane operacje na przykładzie cin, cout
- pojęcie strumienia danych jako warstwy abstrakcji
- tekstowe operacje plikowe jako operacje strumieniowe
- błędy operacji IO

Teaching methods

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

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

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
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	• activity during the classes • an exam - oral, descriptive, test and other • an observation and evaluation of activities during the classes • an observation and evaluation of the student's practical skills • an ongoing monitoring during classes	• 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 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	• an exam - oral, descriptive, test and other • an observation and evaluation of activities during the classes • an observation and evaluation of the student's practical skills	• Lecture • Laboratory
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	• an observation and evaluation of activities during the classes • an observation and evaluation of the student's practical skills	• Laboratory
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	• activity during the classes • an exam - oral, descriptive, test and other • an observation and evaluation of the student's practical skills • 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: 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. Język C. Szkoła programowania. Wydanie VI, Stephen Prata, Helion 2016
2. Podstawy języka C++, Stanley B. Lippman, WNT 2003

Further reading

Internet

Notes

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.

Modified by dr inż. Gerard Bryś (last modification: 16-04-2020 10:10)

Generated automatically from SylabUZ computer system