

Computer Programming 2 - course description

General information	
Course name	Computer Programming 2
Course ID	11.3-WK-liEP-PK2-W-S14_pNadGen3JVWA
Faculty	Faculty of Mathematics, Computer Science and Econometrics
Field of study	Informatics and Econometrics
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	dr Florian Fabiś

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	30	2	-	-	Exam
Laboratory	30	2	-	-	Credit with grade

Aim of the course

Opanowanie umiejętności pisania programów z wykorzystaniem złożonych struktur danych. Umiejętność implementacji w programach algorytmów sortowania, selekcji i wyszukiwania. Znajomość w zakresie podstawowym problemu NP-zupełności (klasy P, NP, NP-zupełne). Zapoznanie studenta z podstawowymi pojęciami programowania zorientowanego obiektowego.

Prerequisites

Programowanie komputerów 1.

Scope

Wykład

- Programowanie strukturalne w języku C++.** Struktury w C++. Zmienne dynamiczne: definicje, procedury tworzenia i usuwania zmiennych dynamicznych, operacje określone na elementach typu wskaźnikowego. Dynamiczne struktury danych: listy, stosy, kolejki. Podstawowe operacje na plikach w C++. Kontenery w C++
- Paradygmat programowania obiektowego.**
- Metody konstruowania efektywnych algorytmów.** Rekurencja, metoda „dziel i zwyciężaj”, algorytmy zachłanne.
- Podstawowe algorytmy wyszukiwania.** Wyszukiwanie liniowe, binarne, interpolacyjne. Samoorganizujące się pliki.
- Podstawowe algorytmy sortowania i selekcji.**
- Klasy złożoności obliczeniowej problemów.** Problemy decyzyjne i optymalizacyjne. Modele obliczeń: deterministyczna i niedeterministyczna maszyna Turinga. Klasy złożoności obliczeniowej problemów decyzyjnych: P, NP, NP-zupełne.

Laboratorium

- Pisanie i uruchamianie programów zawierających tablice struktur.
- Przetwarzanie plików.
- Stosowanie w programach zmiennych dynamicznych i tablic dynamicznych.
- Dynamiczne struktury danych (stosy, kolejki, listy).
- Pisanie programu będącego prostą bazą danych.

Teaching methods

Wykład: wykład problemowy.

Laboratorium: ćwiczenia laboratoryjne w pracowni komputerowej – pisanie i uruchamianie samo-dzielnie napisanych programów w C++, na zadane przez prowadzącego tematy, analiza tych programów i analiza stosowanych algorytmów. Nad niektórymi programami studenci będą pracowali w grupach 2–3 osobowych. Na ostatnich trzech ćwiczeniach każdy student pracuje nad samodzielnie zaprojektowaną aplikacją będącą prostą bazą danych, zawierającą wszystkie zdobyte na zajęciach umiejętności. Do aplikacji musi być dołączona dokumentacja zgodna z zadaną specyfikacją.

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
Student zna podstawowe algorytmy sortowania i potrafi określić ich złożoność.	K_W05	an exam - oral, descriptive, test and other	Lecture

Outcome description	Outcome symbols	Methods of verification	The class form
Student potrafi modelować fragment rzeczywistości przez definiowanie i korzystanie ze złożonych struktur danych.	• K_U28	• a discussion • an ongoing monitoring during classes	• Laboratory
Student wie jak korzystać ze złożonych konstrukcji wybranego języka programowania do rozwiązania zadań.	• K_W10	• a quiz • an exam - oral, descriptive, test and other	• Lecture • Laboratory
Student potrafi wykonywać podstawowe operacje na strukturach dynamicznych.	• K_U20	• a quiz • an ongoing monitoring during classes	• Laboratory
Student dobrze zna podstawowe pojęcia związane z programowaniem obiektowym.	• K_W10	• an exam - oral, descriptive, test and other	• Lecture
Student potrafi tworzyć diagram klas.	• K_U25	• a discussion • a quiz • an exam - oral, descriptive, test and other	• Lecture • Laboratory
Student potrafi definiować proste klasy i korzystać z nich.	• K_U28	• a quiz	• Laboratory
Student potrafi zastosować złożone konstrukcje wybranego języka programowania do rozwiązania zadań.	• K_U25	• a quiz • an exam - oral, descriptive, test and other	• Lecture • Laboratory
Student potrafi przedstawiać algorytmy za pomocą diagramów.	• K_U20	• a quiz • an exam - oral, descriptive, test and other	• Lecture • Laboratory

Assignment conditions

Wykład. Egzamin weryfikujący efekty kształcenia w zakresie wiedzy i umiejętności. Egzamin składa się z dwóch części, pisemnej i ustnej. Warunkiem przystąpienia do części ustnej jest uzyskanie 30% punktów z części pisemnej. Uzyskanie 50% punktów z części pisemnej gwarantuje uzyskanie pozytywnej oceny.

Laboratorium. Ocena końcowa jest wystawiana na podstawie punktów uzyskanych na zajęciach. Punkty uzyskuje się za napisane na zajęciach sprawdziany, przedstawioną na ostatnich zajęciach aplikację, aktywność na zajęciach.

Na ocenę z przedmiotu składa się ocena z laboratorium (50%) oraz ocena z egzaminu (50%). Warunkiem przystąpienia do egzaminu jest pozytywna ocena z laboratorium. Warunkiem zaliczenia przedmiotu jest pozytywna ocena z egzaminu.

Recommended reading

1. L. Banachowski, K. Diks, W. Rytter : Algorytmy i struktury danych, WNT, W-wa 1996.
2. J. Grebosz, Symfonia C++, Wydawnictwo Edition 2000, Kraków, 2010
3. A. Majczak, Programowanie strukturalne i obiektowe, Helion, Gliwice, 2010.
4. A. Stasiewicz, C++. Ćwiczenia praktyczne, Helion, Gliwice, 2008.

Further reading

1. T.H. Cormen, Ch.E. Leiserson, R.L. Rivest: Wprowadzenie do algorytmów, WNT, 1997.
2. P. Wróblewski: Algorytmy, struktury danych i techniki programowania, wyd. II popr., Helion, 2001.

Notes

Modified by dr Robert Dylewski, prof. UZ (last modification: 04-05-2018 19:24)

Generated automatically from SylabUZ computer system