

Principles of programming - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Principles of programming
Kod przedmiotu	11.3-WE-AutP-PrinProgr-Er
Wydział	Wydział Informatyki, Elektrotechniki i Automatyki
Kierunek	Automatyka i robotyka
Profil	ogólnoakademicki
Rodzaj studiów	Program Erasmus
Semestr rozpoczęcia	semestr zimowy 2017/2018

Informacje o przedmiocie	
Semestr	1
Liczba punktów ECTS do zdobycia	5
Typ przedmiotu	obowiązkowy
Język nauczania	angielski
Sylabus opracował	dr inż. Grzegorz Łabiak

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	30	2	-	-	Egzamin
Laboratorium	30	2	-	-	Zaliczenie na ocenę

Cel przedmiotu

The goal of the subject is to teach programming in C language. In the first place the learning consists in teaching syntax and semantics of C language. Next, students are instructed how to implement simple computational algorithms. Apart from that some elements of computational complexity are introduced, which allows to asses computational cost of implemented solutions.

Wymagania wstępne

fundamentals of mathematics

Zakres tematyczny

Introductory information: a program and its components

Programming environment. Source files. Compilation. Basic elements of program and its structure. Main function. Functions and procedures.

Basic types. Variables. In/out operations. Operators.

Iterative loops: for, while, do-while

Decision instructions.

Creation of own Functions.

Arrays and character strings.

Structures.

Pointers.

Sorting algorithms: bubble sort, selection sort, quick sort.

Metody kształcenia

Lecture, laboratory exercises

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Student is able to implement typical algorithms, such as sorting, searching, etc.		kolokwium	- Wykład - Laboratorium
Student is able to define real world problems and present them as computational problems, critically analyse solutions and assess computational complexity		- bieżąca kontrola na zajęciach - sprawdzian	Laboratorium
Student knows the difference between structural and object oriented programming		bieżąca kontrola na zajęciach	Laboratorium

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Student is able to solve algorithmic problem, implement it in C language, run and test it.		- bieżąca kontrola na zajęciach - sprawdzian	Laboratorium

Warunki zaliczenia

Lecture - exam - in order to get a credit it is necessary to pass all of the required tests (oral or written)

Laboratory - the main condition to get a pass are sufficient marks for all exercises and tests conducted during the semester

Calculation of the final Grade: lecture 50% + laboratory 50%

Literatura podstawowa

1. Kernighan B. W., Ritchie D. M.: Język Ansi C, WNT, Warszawa, 1994.
2. Sielicki A.: Laboratorium programowania w języku Pascal, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 1994

Literatura uzupełniająca

1. Aho A. V., Hopcroft J. E., Ullman J. D.: Projektowanie i analiza algorytmów, Helion, Warszawa, 2003.
2. Banachowski L., Diks K., Rytter W.: Algorytmy i struktury danych, WNT Warszawa, 2001.
3. Roszkowski J.: Analiza i projektowanie strukturalne, Helion, Gliwice, 2002.
4. Wirth N.: Algorytmy + struktury danych = programy, WNT, Warszawa, 1989.

Uwagi

Zmodyfikowane przez dr inż. Grzegorz Łabiak (ostatnia modyfikacja: 09-05-2017 14:48)

Wygenerowano automatycznie z systemu SyllabUZ