

Programming languages - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Programming languages
Kod przedmiotu	11.3-WE-ELEKTP-ProLan-Er
Wydział	Wydział Informatyki, Elektrotechniki i Automatyki
Kierunek	Elektrotechnika
Profil	ogólnoakademicki
Rodzaj studiów	Program Erasmus pierwszego stopnia
Semestr rozpoczęcia	semestr zimowy 2017/2018

Informacje o przedmiocie	
Semestr	2
Liczba punktów ECTS do zdobycia	3
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	15	1	-	-	Zaliczenie na ocenę
Laboratorium	30	2	-	-	Zaliczenie na ocenę

Cel przedmiotu

The goal of the subject is to teach programming in C/C++ luguages. In the first place the learning consists in teaching syntax and semantics of C language and some selected C++ language structures. Next, students are instructed how to implement simple computational algorithms. Apart from that some elements of object oriented programming, such as classes are introduced, in order students to be able to implement calcualtions of electrical systems.

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.
- Pointers.
- Sorting algorithms: bubble sort, selection sort, quick sort.
- Classes, objects, inheritance, operator overloading.
- Complex numbers.

Metody kształcenia

Lecture, laboratory exercises

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektówMetody 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
Student is able to define real world problemes and present them as computational problmes, critially 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
Studen is able to implement typical algorithms, such as sorting, searching, etc.		• kolokwium	• Wykład • 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.
3. Stroustrup B.: Język C++, WNT 2002

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 hab. inż. Radosław Kłosiński, prof. UZ (ostatnia modyfikacja: 02-05-2017 18:32)

Wygenerowano automatycznie z systemu SylabUZ