

Nazwa zajęć: Podstawy programowania		The basics of programming	
Kierunek: Informatyka			Obowiązuje od roku ak. 2019/2020
Poziom: I st. inżynierski	Profil: praktyczny	Grupa zajęć: Kierunkowe	
Semestr: I	Forma zaliczenia: E - egzamin	Punkty ECTS: 6	Zajęcia do wyboru: Nie
			Język zajęć: polski

Forma zajęć i liczba godzin na studiach stacjonarnych i niestacjonarnych:

Wykład 30 / 16	Ćwiczenia 30 / 16	Suma godzin: 60 / 32
--------------------------	-----------------------------	--------------------------------

Specjalność:

Nazwiska osób odpowiedzialnych za zajęcia:
dr Piotr Sługocki, dr inż. Włodzimierz Malesa

Opis zajęć:
W ramach przedmiotu student uzyskuje umiejętność czytania ze zrozumieniem programów zapisanych w wybranym języku programowania imperatywnego. Realizowana jest nauka symbolicznego wykonywania prostych programów celem ich weryfikacji. Pisanie i uruchamianie prostych programów o rozmiarze rzędu 100 wierszy kodu
Wykład prowadzony jest metodą podającą z elementami metody praktycznej. Rozszerzenia problemów poruszanych na wykładzie dokonuje się metodą opiszową, w celu zapoznania słuchaczy z nowymi zagadnieniami. Analiza praktycznych zagadnień z dziedziny podstaw programowania jest przekazywana metodą podającą i stanowi potwierdzenie omawianej teorii.

Cele dydaktyczne:

Uzyskanie przez studenta umiejętności czytania ze zrozumieniem programów zapisanych w wybranym języku programowania imperatywnego. Nauka symbolicznego wykonywania prostych programów celem ich weryfikacji. Pisanie i uruchamianie prostych programów o rozmiarze rzędu 100 wierszy kodu
Uzyskanie ogólnej wiedzy z zakresu podstaw programowania z zastosowaniem wybranego języka formalnego, zasad budowy, testowania i usprawniania prostych algorytmów i ich implementacji w wybranym języku formalnym.
Uzyskanie umiejętności w zakresie wyszukiwania w literaturze i źródłach elektronicznych informacji na temat rozwiązań prostych problemów algorytmicznych oraz umiejętności testowania i usprawniania prostych programów zapisanych w wybranym języku formalnym, interpretowania uzyskanych wyników i formułowania wniosków.
Uzyskanie kompetencji w określaniu sposobów postępowania w trakcie wykonywania prostych programów i ich weryfikacji

Metody dydaktyczne:

MP1 wykład informacyjny
MC1 ćwiczenie praktyczne

Metody oceniania:

MO1 test kontrolny
MO2 praca projektowa

Wykład

W1	Pojęcie algorytmu, reprezentacja algorytmu, znajdowanie algorytmu.
W2	Podstawowe konstrukcje programistyczne.
W3	Pojęcie typu danych: proste typy danych, standardowe typy proste, typy okrojone, tablice, rekordy, zbiory, pliki.
W4	Podstawowe operacje na wybranych typach danych.
W5	Implementacje algorytmów w wybranych językach programowania
W6	Dynamiczny przydział pamięci.
W7	Rekurencja i jej implementacja w wybranych językach wysokiego poziomu.
W8	Metody weryfikacji poprawności programów.

Ćwiczenia

C1	Reprezentacja algorytmu, znajdowanie algorytmu oraz zapis formalny z wykorzystaniem wybranych języków programowania wysokiego poziomu.
C2	Implementacja podstawowych konstrukcji programistycznych w wybranych językach programowania.
C3	Zapis formalny wybranych typów danych: proste typy danych, standardowe typy proste, typy okrojone, tablice, rekordy, zbiory, pliki.
C4	Podstawowe operacje na wybranych typach danych.
C5	Implementacje przykładowych algorytmów w wybranych językach programowania.
C6	Zadania z wykorzystaniem dynamicznego przydziału pamięci.
C7	Implementacja przykładowych algorytmów rekurencyjnych w wybranych językach wysokiego poziomu.
C8	Weryfikacja poprawności wybranych programów zrealizowanych w ramach ćwiczeń.

Literatura podstawowa

1 Banachowski L., Diks K., Rytter, W.: Algorytmy i struktury danych, WNT 2001
2 Cormen T.H., Leiserson C. E., Rivest R.L.: Wprowadzenie do algorytmów, WNT 2001
3 Prata S., Język C++. Szkoła programowania, Helion 2012.

Literatura uzupełniająca

1 Aho A.V., Hopcroft J.E., Ullman J.D.: Algorytmy i struktury danych, Helion 2003
2 Owczarek M., Microsoft Visual C++ 2008. Praktyczne przykłady, Helion 2006
3 Felleisen M., Findler R., Flatt M., Krishnamurthi S., Projektowanie oprogramowania. Wstęp do programowania i techniki komputerowej, Helion 2003,
4 Wirth N.: Algorytmy + struktury danych = programy, WNT, 1980,
5 Wróblewski P.: Algorytmy, struktury danych i techniki programowania, Helion 1997.
6 Fortuna Z., Macukow B., Wąsowski J.: Metody numeryczne, WNT 1993,

Warunki zaliczenia

Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnej oceny z testu sprawdzającego wiedzę teoretyczną (min. 51% poprawnych odpowiedzi) oraz wykonania zadania praktycznego na ostatnich zajęciach ćwiczeń laboratoryjnych.

Przykłady pytań zaliczeniowych

Wymienić cechy algorytmu i opisać jedną z wybranych
Przedstawić graficznie przykład algorytmu rozgałęzionego
Zapisać w postaci przykładu rekordowy typ danych
Przedstawić przykład sortowania „przez wstawianie”
Wymienić podstawowe typy proste
Zapisać w postaci przykładu rekordowy typ danych

Obciążenie pracą studenta*Studia stacjonarne/niestacjonarne*

Forma pracy studenta	Wykład		Ćwiczenia		Suma	
Zajęcia z bezpośrednim udziałem nauczyciela	30 g	16 g	30 g	16 g	60 g	32 g
Zapoznanie się z literaturą przedmiotu	10 g	16 g	20 g	16 g	30 g	32 g
Przygotowanie się do zajęć	10 g	16 g	10 g	16 g	20 g	32 g
Przygotowanie się do kolokwium	10 g	16 g	10 g	16 g	20 g	32 g
Realizacja zadanych ćwiczeń i zadań			20 g	22 g	20 g	22 g
Przygotowanie sprawozdania z ćwiczeń						
Przygotowanie projektu / pracy						
Przygotowanie się i udział w egzaminie						
	60 g	64 g	90 g	86 g	150 g	150 g

Efekty uczenia się	KEK	Treści kształcenia	Metody dydaktyczne	M. oceniania
zna podstawowe zagadnienia w dziedzinie programowania	K_W04	W1-W8	MP1	MO1
zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu praktycznych zadań inżynierskich z zakresu programowania	K_W05	W1-W8	MP1	MO1
potrafi dobrać właściwe metody i narzędzia oraz umie posługiwać się technikami informacyjno-komunikacyjnymi do realizacji zadań typowych dla działalności inżynierskiej w dziedzinie programowania	K_U04	W1-W8 C1-C8	MP1, MC1	MO1, MO2
potrafi wykorzystywać posiadaną wiedzę z zakresu informatyki poprzez umiejętność formułowania i rozwiązywania problemów praktycznych z algorytmiki	K_U05	W1-W8 C1-C8	MP1, MC1	MO1, MO2
gotów jest do krytycznej oceny posiadanej wiedzy i odbieranych treści w dziedzinie programowania	K_K01	W1-W8 C1-C8	MP1, MC1	MO1, MO2
gotów jest do rozwiązywania problemów poznawczych i praktycznych oraz zasięgania opinii ekspertów w przypadku trudności z samodzielnym rozwiązywaniem problemów w dziedzinie programowania	K_K02	W1-W8 C1-C8	MP1, MC1	MO1, MO2