

# Podstawy programowania - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Podstawy programowania
Kod przedmiotu	11.3-WE-ELEKTP-PodstProgr
Wydział	<a href="#">Wydział Informatyki, Elektrotechniki i Automatyki</a>
Kierunek	Elektrotechnika
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. inżyniera
Semestr rozpoczęcia	semestr zimowy 2018/2019

Informacje o przedmiocie	
Semestr	3
Liczba punktów ECTS do zdobycia	4
Typ przedmiotu	obowiązkowy
Język nauczania	polski
Sylabus opracował	<ul style="list-style-type: none"><li>• dr inż. Paweł Majdzik</li><li>• dr inż. Mirosław Koziół</li></ul>

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	9	0,6	Zaliczenie na ocenę
Laboratorium	30	2	18	1,2	Zaliczenie na ocenę

## Cel przedmiotu

Celem przedmiotu jest nauka programowania w języku C. Nauka polega na zapoznaniu się ze składnią i semantyką języka C oraz na kształtowaniu umiejętności implementowania prostych algorytmów.

## Wymagania wstępne

Zagadnienia z matematyki z zakresu szkoły średniej.

## Zakres tematyczny

Wstęp do programowania w języku C: Systemy liczbowe (dziesiętny, binarny i szesnastkowy). Podstawowe elementy i budowa programu. Funkcja główna. Funkcje i procedury. Środowisko programistyczne. Pliki źródłowe.

Kompilacja i linkowanie programu.

Stałe i zmienne: Podstawowe typy danych. Deklaracje zmiennych i ich zasięg. Stałe i preprocesor. Tablice.

Operatory, wyrażenia i instrukcje: Operatory arytmetyczne i logiczne. Manipulowanie bitami (bitowe operatory logiczne). Konwersja typów (operator rzutowania). Instrukcje sterujące (pętle, rozgałęzienia i skoki).

Operacje wejścia/wyjścia: Łańcuchy znakowe. Funkcje getchar, putchar i funkcje formatujące printf i scanf. Operacje na ciągach znaków.

Funkcje: Tworzenie i korzystanie z funkcji. Argumenty funkcji. Zwracanie wartości przy pomocy instrukcji return.

Wskaźniki: Uzyskiwanie adresów (operator &). Wskaźniki (operator dereferencji \*). Deklarowanie wskaźników. Dynamiczne przydzielanie pamięci. Korzystanie z argumentów wskaźnikowych w przypadku argumentów funkcji.

Struktury: Deklaracja struktury. Definiowanie własnych typów danych (typedef). Definiowanie zmiennej strukturalnej. Uzyskiwanie dostępu do składników struktury.

## Metody kształcenia

**wykład:** wykład konwencjonalny, wykład problemowy, dyskusja

**laboratorium:** ćwiczenia laboratoryjne

## Efekty kształcenia i metody weryfikacji osiągnięcia efektów kształcenia

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
-------------	-----------------	--------------------	-------------

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Potrąfi opracowane rozwiązanie problemu algorytmicznego zaimplementować w języku C, uruchomić i testować powstały program komputerowy.	<ul style="list-style-type: none"> <li>• <a href="#">K_W09</a></li> <li>• <a href="#">K_U11</a></li> </ul>	<ul style="list-style-type: none"> <li>• bieżąca kontrola na zajęciach</li> <li>• sprawdzian</li> </ul>	<ul style="list-style-type: none"> <li>• Laboratorium</li> </ul>
Student potrafi formułować problemy świata rzeczywistego jako typowe problemy algorytmiczne, dokonywać ich analizy pod kątem ich rozwiązania i oceny złożoności obliczeniowej.	<ul style="list-style-type: none"> <li>• <a href="#">K_U11</a></li> </ul>	<ul style="list-style-type: none"> <li>• bieżąca kontrola na zajęciach</li> <li>• sprawdzian</li> </ul>	<ul style="list-style-type: none"> <li>• Laboratorium</li> </ul>
Ma potrzebną wiedzę i umiejętności aby implementować typowe algorytmy (np. wyszukiwania, problemy grafowe itp.) w języku C.	<ul style="list-style-type: none"> <li>• <a href="#">K_W09</a></li> <li>• <a href="#">K_U11</a></li> </ul>	<ul style="list-style-type: none"> <li>• bieżąca kontrola na zajęciach</li> <li>• sprawdzian</li> </ul>	<ul style="list-style-type: none"> <li>• Wykład</li> <li>• Laboratorium</li> </ul>

## Warunki zaliczenia

**Wykład** - egzamin - umiejętność rozwiązywania zadań; znajomość składni i semantyki języka C

**Laboratorium** - warunkiem zaliczenia jest uzyskanie pozytywnych ocen ze wszystkich ćwiczeń laboratoryjnych, przewidzianych do realizacji w ramach programu laboratorium

**Składowe oceny końcowej** = wykład: 50% + laboratorium: 50%

## Obciążenie pracą

Obciążenie pracą	Studia stacjonarne (w godz.)	Studia niestacjonarne (w godz.)
Godziny kontaktowe (udział w zajęciach; konsultacjach; egzaminie, itp.)	65	40
Samodzielna praca studenta (przygotowanie do: zajęć, kolokwium, egzaminu; studiowanie literatury przygotowanie: pracy pisemnej, projektu, prezentacji, raportu, wystąpienia; itp.)	35	60
<b>Łącznie</b>	<b>100</b>	<b>100</b>
Punkty ECTS	Studia stacjonarne	Studia niestacjonarne
Zajęcia z udziałem nauczyciela akademickiego	3	2
Zajęcia bez udziału nauczyciela akademickiego	1	2
<b>Łącznie</b>	<b>4</b>	<b>4</b>

## 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 hab. inż. Radosław Kłosiński, prof. UZ (ostatnia modyfikacja: 25-04-2018 23:07)

Wygenerowano automatycznie z systemu SylabUZ