**Nazwa przedmiotu:**

Podstawy programowania (I)

**Koordynator przedmiotu:**

Henryk DOBROWOLSKI

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

PRI

**Semestr nominalny:**

1 / rok ak. 2018/2019

**Liczba punktów ECTS:**

6

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

30 godz. wykładu
15 godz. ćwiczeń
30 godz. przygotowanie do sprawdzianów
30 godz. laboratorium
45 godz. przygotowania do laboratorium
w sumie 150 godz. co daje 6 ECTS

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

30 godz. wykładu
15 godz. ćwiczeń
30 godz. laboratorium
w sumie 75 godz. co daje 3 ECTS

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

15 godz. ćwiczeń
15 godz. przygotowanie do sprawdzianów
30 godz. laboratorium
45 godz. przygotowania do laboratorium
w sumie 105 godz. co daje 4 ECTS

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 15h |
| Laboratorium: | 30h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

brak - przedmiot I semestru

**Limit liczby studentów:**

przedmiot obowiązkowy dla kierunku informatyki

**Cel przedmiotu:**

• Nauczenie podstaw algorytmizacji problemów
• Zapoznanie z techniką programowania strukturalnego
• Wprowadzenie do programowania w językach wysokiego poziomu
• Praktyczna nauka podstaw programowania w oparciu o język C
• Zapoznanie z wybranymi algorytmami i konstrukcjami danych w oparciu o reprezentację składniową języka C (z odniesieniami do C++ i Pascal’a)
• Wprowadzenie do podstawowych pojęć inżynierii oprogramowania – specyfikacja, dokumentacja, testowanie i pielęgnacja oprogramowania

**Treści kształcenia:**

Pojęcia podstawowe (algorytm, program, proces. Reprezentowanie algorytmów. Translacja. Kompilacja i konsolidacja programu. Biblioteki. Środowisko kodowania i uruchamiania programów – edytor, kompilator, debuger).
Podstawy algorytmizacji (Dekompozycja problemu. Podejście strukturalne do projektowania algorytmów. Konstruowanie prostych algorytmów. Iteracja i rekursja. Ocena jakości algorytmu. Testowanie algorytmu; dobór danych wejściowych, badanie przypadków granicznych i wyjątków. Sprawność i złożoność algorytmu).
Języki programowania (Przegląd – od maszynowo zorientowanych do proceduralnych. Syntaktyka i sposoby jej opisu – diagramy syntaktyczne, notacja BNF. Semantyka. Typy instrukcji/operacji – składnia dla języka C. Struktura programu ).
Elementy języka C (Pojęcia podstawowe - deklaracje i definicje; struktury danych i funkcje. Funkcja main. Symbole i komentarze. Liczbowe typy danych. Wyrażenia arytmetyczne i logiczne - operatory; priorytety i łączność. Instrukcje, ich klasyfikacja, składnia i semantyka. Funkcje - budowa, parametry, wartości zwracane itd. Agregaty danych - tablice i struktury. Wejście/wyjście w C)
Przegląd wybranych algorytmów (Elementarne algorytmy iteracyjne. Manipulacje na tekstach, algorytmy sortowania itd. Iteracja a rekurencja. Algorytmy z powrotami. Dynamiczne struktury danych – algorytmy dla list i drzew. Wyszukiwanie)
Elementy wstępu do inżynierii oprogramowania (Paradygmaty programowania. Modele - kaskadowy, spiralny. Testowanie. Cykl życia oprogramowania).

**Metody oceny:**

Na ocenę składają się elementy:
1. oceny z kolokwiów wykładowych (2 kolokwia)
2. ocena z ćwiczeń audytoryjnych
3. oceny z zadań laboratoryjnych (w ciągu semestrui - 4 indywidualne zadania o sukcesywnie wzrastającym stopniu złożoności)

**Egzamin:**

nie

**Literatura:**

Literatura podstawowa
- B.W.Kernighan, D.M.Ritchie Język ANSI C
Do samodzielnej nauki:
- S.Prata Szkoła Programowania – Język C
- H.Schildt Programowanie C (ale tłumaczenie wydania amerykańskiego z 2000 roku – uwzględnia standard C99)
Dla zaawansowanych:
- Aho – Ullman Wykłady z informatyki z przykładami w języku C
- B.W,Kernighan, R.Pike Lekcja programowania

**Witryna www przedmiotu:**

http://studia.elka.pw.edu.pl

**Uwagi:**

Przedmiot jest pierwszym z ciągu poświęconego programowaniu.

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka PRI\_W01:**

Znajomość podstaw strukturalizacji rozwiązywania problemów

Weryfikacja:

Ćwiczenia, zadania laboratoryjne

**Powiązane charakterystyki kierunkowe:** K\_W19

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

**Charakterystyka PRI\_W02:**

Znajomość języka C na poziomie umożliwiającym projektowanie względnie prostych aplikacji

Weryfikacja:

Ćwiczenia, zadania laboratoryjne, kolokwia

**Powiązane charakterystyki kierunkowe:** K\_W19

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

**Charakterystyka PRI\_W03:**

Znajomość podstaw algorytmizacji

Weryfikacja:

Ćwiczenia, zadania laboratoryjne

**Powiązane charakterystyki kierunkowe:** K\_W12

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

**Charakterystyka PRI\_W04:**

Znajomość podstawowych pojęć związanych z cyklem życia oprogramowania

Weryfikacja:

Ćwiczenia

**Powiązane charakterystyki kierunkowe:** K\_W15

**Powiązane charakterystyki obszarowe:** I.P6S\_WG, III.P6S\_WG.o

### Profil ogólnoakademicki - umiejętności

**Charakterystyka PRI\_U01:**

Reprezentowanie algorytmów w formie schematów blokowych / diagramów aktywności

Weryfikacja:

Ćwiczenia, zadania laboratoryjne

**Powiązane charakterystyki kierunkowe:** K\_U13

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, III.P6S\_UW.4.o

**Charakterystyka PRI\_U02:**

Kodowanie algorytmów z zastosowaniem języka C

Weryfikacja:

Zadania laboratoryjne, kolokwia

**Powiązane charakterystyki kierunkowe:** K\_U13

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, III.P6S\_UW.4.o

**Charakterystyka PRI\_U03:**

Projektowanie, kodowanie, uruchamianie i testowanie prostych programów aplikacyjnych (tryb tekstowy) z zastosowaniem języka C

Weryfikacja:

Zadania laboratoryjne, kolokwia

**Powiązane charakterystyki kierunkowe:** K\_UK04

**Powiązane charakterystyki obszarowe:** I.P6S\_UO