**Nazwa przedmiotu:**

Podstawy programowania strukturalnego

**Koordynator przedmiotu:**

dr inż. Władysław Homenda

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

**Semestr nominalny:**

1 / rok ak. 2012/2013

**Liczba punktów ECTS:**

6

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

1. godziny kontaktowe - 80 h; w tym
a. obecność na wykładach – 30 h
b. obecność na ćwiczeniach – 30 h
c. obecność na laboratoriach – 15 h
d. konsultacje – 5 h
2. przygotowanie do zajęć – 85 h, w tym
a. przygotowanie do wykładów – 15 h
b. przygotowanie do ćwiczeń – 20 h
c. dodatkowo przygotowanie do sprawdzianów pisemnych – 20 h
d. przygotowanie do zajęć laboratoryjnych – 30 h
Razem nakład pracy studenta 165 h = 6 pkt. ECTS

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

1. obecność na wykładach – 30 h
2. obecność na ćwiczeniach – 30 h
3. obecność na laboratoriach – 15 h
4. konsultacje z prowadzącymi zajęcia – 5 h
Razem 30+30+15+5 = 80 h, co odpowiada 3 pkt. ECTS

**Język prowadzenia zajęć:**

polski

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

1. obecność na ćwiczeniach – 30 h
2. obecność na laboratoriach – 15 h
3. przygotowanie do ćwiczeń – 20 h
4. przygotowanie do sprawdzianów pisemnych – 20 h
5. przygotowanie do zajęć laboratoryjnych – 30 h
Razem 30+15+20+20+30= 115 h, co odpowiada 4 pkt. ECTS

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

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

**Wymagania wstępne:**

brak

**Limit liczby studentów:**

Bez limitu

**Cel przedmiotu:**

Celem przedmiotu jest nauka podstaw programowania strukturalnego. Przedmiot jest skierowany do osób, które wcześniej nie uczyły się programować. Po ukończeniu kursu studenci powinni wiedzę i umiejętności sformułowane w tabeli efektów kształcenia.

**Treści kształcenia:**

1)     
a)    Informacje wstępne: problem, algorytm.
b)    Systemy pozycyjne: dziesiętny, dwójkowy, szesnastkowy, jedynkowy (liczba n jest zapisana jako ciąg n zer), minus dwójkowy.
c)    Reprezentacja liczb całkowitych w pamięci komputera – system stałopozycyjny uzupełnieniowy.
2)     
a)    Reprezentacja liczb zmiennoprzecinkowych, formaty IBM, IEEE, Microsoft, zakresy liczb zmiennoprzecinkowych.
b)    Wyrażenia arytmetyczne – priorytety operatorów, notacje przedrostkowa i przyrostkowa.
3)     
a)    Prosty model maszyny cyfrowej
b)    Prosty pseudoassembler, algorytm Euklidesa wyznaczania NWD, sito Eratostenesa .
4)     
a)    Języki wysokiego poziomu, C – informacje ogólne.
b)    Struktura programu, funkcje.
c)    Podstawowe typy danych – int, float; modyfikatory: long, short, unsigned.
5)     
a)    Typy i rozmiary danych.
b)    Stałe, zmienne, typy zmiennych.
c)    Funkcje konwersji typów danych.
6)     
a)    Tablice.
b)    Struktury.
c)    Typy złożone (np. tablice struktur).
7)     
a)    Operatory,
b)    Priorytety, konwersje typów.
c)    Wyrażenia – arytmetyczne, logiczne, binarne, rzutowania,
d)    Instrukcje proste -  pusta, podstawienia, skoku
e)    Instrukcje złożone - strukturalne, warunkowe, wyboru, iteracyjne
8)     
a)    Instrukcje – uzupełnienia
b)    Prosty program w C – szybkie obliczanie potęg (przykład na zastosowanie operatorów, wyrażeń, instrukcji warunkowych i iteracyjnych)
c)    Prosty program w C –wyszukiwanie wzorca w tekście (przykład na zastosowanie tablic i instrukcji warunkowych, wyboru i iteracyjnych),
9)     
a)    Podstawowe instrukcje wejścia-wyjścia: getch, putc, scanf, printf, elementy formatowania
b)    Przykłady programów: modyfikacja sita Eratostenesa na tablicach – znajdowanie liczb słabo złożonych,
10) Wskaźniki
a)    Zmienne wskaźnikowe, zmienne wskazywane, dynamiczne zarządzanie pamięcią.
b)    Przykłady - sito Eratostenesa w wersji listy jednokierunkowej – tworzenie listy jednokierunkowej, usuwanie elementów z listy i nawigacja po liście jednokierunkowej,
11) Wskaźniki – cd.
a)    tworzenie uporządkowanej listy dwukierunkowej,
b)    Wskaźniki i tablice, wskaźniki do wskaźników – omówienie.
12) Funkcje i programy – wprowadzenie
a)    Funkcje – sposoby przekazywania parametrów: przez wartość, przez adres, przez funkcję.
b)    Funkcje i programy – przykłady :
i)      Program obliczania exp(A), gdzie A – macierz nxn
ii)     Program wyznaczania całki
13) Funkcje i programy – przykłady, cd.
a)    Budowa drzewa binarnego,
b)    Przeglądanie drzewa binarnego: prefiksowe, postfiksowe, infiksowe,
14) Funkcje i programy – przykłady, cd.
a)    Rekurencja, omówienie, przykład: wieże Hanoi, gdzie nie należy stosować rekurencji – obliczanie silni
b)    przesłanianie,
c)    efekty uboczne.
15) Zaliczenie

**Metody oceny:**

Zaliczenie ćwiczeń - dwa kolokwia w semestrze. Zaliczenie laboratoriów na podstawie programów pisanych w laboratorium. Zaliczenie przedmiotu: wymagane zaliczenie ćwiczeń i laboratoriów. Łączną ocenę punktową przelicza się na stopnie według poniższych zasad:
b)  3.5 jeżeli uzyskali od 61 do 70  pkt.
c)  4.0 jeżeli uzyskali od 71 do 80  pkt.
d)  4.5 jeżeli uzyskali  od 81 do 90  pkt.
e)  5.0 jeżeli uzyskali powyżej 90  pkt.

**Egzamin:**

nie

**Literatura:**

Kerningham, Ritchie, Język C
Wirth, Algorytmy + struktury danych = programy

**Witryna www przedmiotu:**

brak

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt W01:**

Zna podstawy działania komputera (procesor, rejestry pamięć, język maszynowy i asembler), zna metody reprezentacji danych numerycznych (zapis stałopozycyjny, kod uzupełnieniowy do 2, zapis zmiennopozycyjny) i tekstowych w pamięci komputera

Weryfikacja:

aktywny udział w ćwiczeniach, sprawdzian pisemny,

**Powiązane efekty kierunkowe:** K\_W01, K\_W03

**Powiązane efekty obszarowe:** T1A\_W01, T1A\_W02, T1A\_W07

**Efekt W02:**

Zna pojęcie problemu i algorytmu i metody formułowania algorytmów (schematy blokowe, opisy słowne), zna podstawowe konstrukcje programistyczne (wyrażenia, operacja przypisania, konstrukcje iteracyjne i warunkowe)

Weryfikacja:

aktywny udział w ćwiczeniach, sprawdzian pisemny,

**Powiązane efekty kierunkowe:** K\_W01, K\_W03, K\_W10

**Powiązane efekty obszarowe:** T1A\_W01, T1A\_W02, T1A\_W07, T1A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt U01:**

Potrafi sformułować proste algorytmy w postaci schematów blokowych i opisów słownych

Weryfikacja:

aktywny udział w ćwiczeniach, sprawdzian pisemny,

**Powiązane efekty kierunkowe:** K\_U01, K\_U09, K\_U11, K\_U29

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U09, T1A\_U09, T1A\_U14, T1A\_U15, T1A\_U15

**Efekt U02:**

Potrafi zastosować podstawowe typy i struktury danych (skalarne, tablicowe, strukturalne) oraz podstawowe konstrukcje programistyczne (operatory, wyrażenia, instrukcje proste i złożone, instrukcje iteracyjne, instrukcje warunkowe, instrukcje wej/wyj) do konstrukcji prostych programów w języku programowania C

Weryfikacja:

aktywny udział w ćwiczeniach, sprawdzian pisemny, program w języku C

**Powiązane efekty kierunkowe:** K\_U09, K\_U11, K\_U29

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U09, T1A\_U14, T1A\_U15, T1A\_U15

**Efekt U03:**

Potrafi wydzielić logiczne i funkcjonalne fragmenty programu w postaci procedur (funkcji) języka C, potrafi stosować metody komunikacji między procedurami języka C (parametry formalne, zmienne globalne).

Weryfikacja:

aktywny udział w ćwiczeniach, sprawdzian pisemny, program w języku C

**Powiązane efekty kierunkowe:** K\_U09, K\_U11, K\_U29

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U09, T1A\_U14, T1A\_U15, T1A\_U15

**Efekt U04:**

Potrafi korzystać z zaawansowanych technik programistycznych: dynamiczne zarządzanie pamięcią, dynamiczne struktury danych (tablice o zmiennych rozmiarach, listy), rekurencja.

Weryfikacja:

aktywny udział w ćwiczeniach, sprawdzian pisemny, program w języku C

**Powiązane efekty kierunkowe:** K\_U09, K\_U11, K\_U29

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U09, T1A\_U14, T1A\_U15, T1A\_U15

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt K01:**

Rozróżnia między wiedzą uniwersalną w dziedzinie programowania a umiejętnościami technicznymi zależnymi od stosowanych rozwiązań technologicznych. Rozumie potrzebę doskonalenia w zakresie wiedzy uniwersalnej i konieczność ciągłego uzupełniania umiejętności technicznych.

Weryfikacja:

udział w dyskusji,

**Powiązane efekty kierunkowe:** K\_K01, K\_K02, K\_K04

**Powiązane efekty obszarowe:** T1A\_K01, T1A\_K01, T1A\_K02, T1A\_K05