**Nazwa przedmiotu:**

Zasady programowania strukturalnego I

**Koordynator przedmiotu:**

prof. nzw. dr hab. inż. Barbara Putz

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Automatyka i Robotyka

**Grupa przedmiotów:**

Obowiązkowe

**Kod przedmiotu:**

ZAP

**Semestr nominalny:**

1 / rok ak. 2013/2014

**Liczba punktów ECTS:**

5

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

5 ECTS = 140 godz., w tym: wykład 15, zajęcia laboratoryjne 30, studia literaturowe 20, przygotowanie do zajęć laboratoryjnych 15, rozwiązywanie zadań domowych 30, przygotowanie do sprawdzianów 10, przygotowanie do egzaminu i uczestniczenie w nim 20.

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

2 ECTS = 47 godz., w tym: wykład 15, zajęcia laboratoryjne 30, egzamin 2.

**Język prowadzenia zajęć:**

polski

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

3 ECTS= 85 godz., w tym: zajęcia laboratoryjne 30, przygotowanie do zajęć laboratoryjnych 15, rozwiązywanie zadań domowych 30, przygotowanie do sprawdzianów 10.

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

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

**Wymagania wstępne:**

Z wyjątkiem matematyki w zakresie szkoły średniej, przedmiot nie wymaga od studentów wiadomości z innych przedmiotów.

**Limit liczby studentów:**

15 osób w grupie laboratoryjnej

**Cel przedmiotu:**

Nauka logicznego, algorytmicznego myślenia i gruntowne wprowadzenie do programowania strukturalnego, wraz z prezentacją zasad konstruowania algorytmów i struktur danych w procesie tworzenia złożonych programów. Nabycie praktycznych umiejętności programowania, ułatwiających badanie i rozwiązywanie problemów inżynierskich, z wykorzystaniem strukturalnych konstrukcji języka C/C++, charakterystycznych dla języków wyższego poziomu. Uzyskana wiedza i umiejętności będą przydatne w całym dalszym toku studiów, ułatwią też szybkie przejście do programowania w Matlabie, językach skryptowych (np. Python czy PHP) oraz do nauki programowania zorientowanego obiektowo (w C++ i Javie).

**Treści kształcenia:**

===WYKŁAD===
CZĘŚĆ 1. Podstawowe pojęcia, algorytm i program. Proste instrukcje i wyrażenia, instrukcje warunkowe. Pętle while, do-while i pętla for. Tablice jednowymiarowe i algorytmy ich sortowania. Tablice dwuwymiarowe. Uzupełnienia: zapis wyrażeń, formaty wydruku. Struktury i rekordy. Definiowanie plików tekstowych, operacje wejścia - wyjścia.
CZĘŚĆ 2. Funkcje - definicja, wywołanie, wiązanie parametrów. Zasięg zmiennych. Rekurencja - zasada działania, przykłady algorytmów. Zmienne dynamiczne i wskaźniki. Tablice dynamiczne jedno- i dwuwymiarowe. Listy dynamiczne jednokierunkowe: tworzenie, wstawianie i usuwanie elementów.
CZĘŚĆ 3. Drzewa binarne i drzewa BST. Drzewa zrównoważone (AVL i czerwono-czarne), kopce, B-drzewa, złożoność obliczeniowa. Grafy i algorytmy grafowe: Floyda, Dijkstry i Kruskala. Przegląd rodzajów i zasad konstruowania algorytmów. Modułowa budowa programów. Podstawowe pojęcia programowania obiektowego jako wprowadzenie do aplikacji okienkowych. Przykłady zadań egzaminacyjnych.

===LABORATORIUM===
Pierwsze programy w wersji konsolowej. Środowisko kompilatora (Qt Creator): edycja, kompilacja i wykonanie programu. Pętle i zaawansowane rozgałęzienia. Tablice - operacje na wektorach i tablicach dwuwymiarowych. Struktury, rekordy i tablice rekordów. Pliki tekstowe, operacje wejścia - wyjścia. Funkcje - deklaracja i wywołanie, parametry formalne i aktualne. Wskaźniki i tablice dynamiczne. Listy jednokierunkowe i podstawowe operacje na nich. Funkcje rekurencyjne.
SPRAWDZIAN NR 1 - opracowanie w rękopisie dwóch prostych programów (if-y, pętle, tablice, rekordy) - 45 min, opracowanie i uruchomienie trzeciego programu z tego zakresu na komputerze - 45 min.
SPRAWDZIAN NR 2 - opracowanie w rękopisie dwóch programów z użyciem funkcji i list jednokierunkowych - 45 min, opracowanie i uruchomienie trzeciego programu z tego zakresu na komputerze - 45 min.

**Metody oceny:**

Wykład - egzamin złożony z części zadaniowej (18 pkt - opracowanie w rękopisie 5 zadań wymagających napisania funkcji lub całych programów) oraz części testowej (8 pkt) z zakresu algorytmów i struktur danych.
Laboratorium - dwa sprawdziany (24 pkt) polegające na samodzielnym napisaniu i uruchomieniu programów w trakcie zajęć.

**Egzamin:**

tak

**Literatura:**

1 Paweł Wnuk, Barbara Putz: Programowanie. Wersja w języku C/C++. OKNO PW, Warszawa 2005, dostępny online w wydziałowym systemie SKS.
2 Barbara Putz, Andrzej Putz jr, Paweł Wnuk: Algorytmy i Struktury Danych. OKNO PW, Warszawa 2008, dostępny online w wydziałowym systemie SKS.
3 Stephen Prata: Język C++. Szkoła programowania. Wydanie V. Helion 2006.
4.Jerzy Grębosz: Symfonia C++ standard. Tom I. Edition 2000 Kraków, 2005-2008
5 Niklaus Wirth: Algorytmy+struktury danych=programy. WNT 2002.
6 Piotr Wróblewski: Algorytmy, struktury danych i techniki programowania. Wyd. IV, Helion 2010.
7 Thomas H. Cormen et al.: Wprowadzenie do algorytmów. WNT 2007, PWN 2012 (nowe wydanie).

**Witryna www przedmiotu:**

http://iair.mchtr.pw.edu.pl/studenci/witryna/index.php (wydziałowy system SKS, wymaga logowania)

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt ZAP1\_W01 :**

ma uporządkowaną wiedzę ogólną obejmującą kluczowe zagadnienia z zakresu konstrukcji języka strukturalnego i programowania strukturalnego w języku C/C++

Weryfikacja:

sprawdzian 1, sprawdzian 2, egzamin

**Powiązane efekty kierunkowe:** K\_W04, K\_W05

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W02, T1A\_W04

**Efekt ZAP1\_W02:**

ma uporządkowaną wiedzę ogólną obejmującą zagadnienia konstruowania algorytmów dla prostych zadań przetwarzania danych

Weryfikacja:

sprawdzian 1, sprawdzian 2, egzamin

**Powiązane efekty kierunkowe:** K\_W04, K\_W05

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W02, T1A\_W04

**Efekt ZAP1\_W03:**

ma uporządkowaną wiedzę ogólną z zakresu klasyfikacji algorytmów i doboru struktur danych

Weryfikacja:

egzamin

**Powiązane efekty kierunkowe:** K\_W04, K\_W05

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W02, T1A\_W04

### Profil ogólnoakademicki - umiejętności

**Efekt ZAP1\_U01:**

ma praktyczną umiejętność opracowania algorytmu i wynikającego stąd programu strukturalnego w języku C/C++ (z wykorzystaniem struktur dynamicznych włącznie) dla prostego zadania programistycznego

Weryfikacja:

sprawdzian 1, sprawdzian 2, egzamin

**Powiązane efekty kierunkowe:** K\_U01, K\_U05, K\_U09

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U05, T1A\_U09

**Efekt ZAP1\_U02:**

ma praktyczną umiejętność napisania i uruchomienia w trakcie 45 min. zajęć programu w środowisku C/C++ na podstawie otrzymanego zadania

Weryfikacja:

sprawdzian 1, sprawdzian 2

**Powiązane efekty kierunkowe:** K\_U01, K\_U05, K\_U09

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U05, T1A\_U09

**Efekt ZAP1\_U03:**

ma umiejętność posługiwania się kompilatorem i debuggerem

Weryfikacja:

sprawdzian 1, sprawdzian 2

**Powiązane efekty kierunkowe:** K\_U01, K\_U05, K\_U09

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U05, T1A\_U09

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt ZAP1\_K01:**

ma nawyk ustawicznego kształcenia się i wyszukiwania nowych informacji, aby radzić sobie z rozwiązywaniem nowych, nietypowych zadań

Weryfikacja:

sprawdzian 1, sprawdzian 2, egzamin

**Powiązane efekty kierunkowe:** K\_K01

**Powiązane efekty obszarowe:** T1A\_K01