**Nazwa przedmiotu:**

Podstawy programowania (M)

**Koordynator przedmiotu:**

Jarosław ARABAS

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Elektronika

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

PRM

**Semestr nominalny:**

1 / rok ak. 2015/2016

**Liczba punktów ECTS:**

4

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

120

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

**Język prowadzenia zajęć:**

polski

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

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

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

**Wymagania wstępne:**

Podstawowe umiejętności w zakresie obsługi komputera. Umiejętność abstrakcyjnego myślenia. Umiejętność dostosowania się do regulaminu przedmiotu.

**Limit liczby studentów:**

120

**Cel przedmiotu:**

Celem przedmiotu jest zapoznanie studentów z paradygmatem programowania strukturalnego oraz zaznajomienie z językiem C w stopniu wystarczającym do samodzielnego pisania średnio złożonych programów oraz współpracy w zespole programistów.

**Treści kształcenia:**

Treść wykładu

Podstawowe pojęcia: algorytm, program, informatyka, programowanie. Narzędzia wspomagające programowanie. Pojęcie typu i wartości. Pierwszy program w języku C.

System operacyjny. System plików. Zasoby i ich ochrona. Procesy. Praca zdalna. Narzędzia ftp, telnet. Poczta elektroniczna.

Stałe: typ i wartość. Zmienne: typ, wartość, nazwa. Typy podstawowe języka C. Niejawne rzutowanie typów. Pojecie deklaracji i definicji zmiennej. Wyrażenia i instrukcje. Operatory arytmetyczne i przyrównania. Operator podstawienia. L-wartości.

Funkcja jako nazwany blok kodu. Pojęcie zmiennej lokalnej i globalnej. Przysłanianie zmiennych. Parametry formalne i aktualne: Instrukcja return. Niejawne rzutowania podczas uzgadniania parametrów formalnych i aktualnych oraz przy ustalaniu wartości zwracanej. Funkcje typu void.
Bloki kodu. Instrukcja if/else. Instrukcja switch - wielowejściowy blok kodu. Instrukcja break. Pojęcie iteracji. Instrukcje pętli: for, while, do/while. Instrukcja continue. Operatory post- i pre- inkrementacji/dekrementacji. Wyodrębnianie atomów leksykalnych. Priorytety i łączność operatorów. Operatory bitowe i ich relacja z opeatorami logicznymi.

Tablica jako ponumerowany zbiór wartości tego samego typu. Pojęcie wskaźnika. Tablice a wskaźniki. "Typ" łańcuchowy. Operatory referencji i dereferencji.

Arytmetyka wskaźników. Rzutowanie wskaźników. Przekazywanie argumentu funkcji przez wskazanie. Przekazywanie tablicy do funkcji. Funkcja main z argumentami wywołania. Zasięg i "żywotność" zmiennych. Klasy zmiennych: static, auto, register, zmienne globalne. Deklaracja zmiennej jako extern.

Dynamiczna alokacja pamięci. Funkcje malloc i free. Patologie z użyciem dynamicznego przydziału pamięci: lokalna zmienna wskaźnikowa, podstawienie nowej wartości zmiennej wskaźnikowej Sterta. Stos (kolejka LIFO). Przykład alokacji tablicy wielowymiarowej. Struktura (struct) jako zbiór wartości z dostępem przez nazwę.

Wejście i wyjście w C. Pojęcie strumienia. Strumień jako kolejka FIFO. Demony. Strumienie plikowe. Tryby otwarcia: tekstowy i binarny. Synchronizacja zawartości strumienia i zawartości pliku. Preprocesor.

Elementy algorytmiki na podstawie metod sortowania.

Rekurencja: zastosowanie, sposób działania, ślad w pamięci operacyjnej.

Przekazywanie argumentów przez zmienną w innych językach programowania. Organizacja pamięci na stosie. Mechanizmy wywołań funkcji.
Zakres laboratorium

Laboratorium wprowadza umiejętność samodzielnego pisania, uruchamiania, testowania i dokumentacji oprogramowania..

 System plików, procesy, sposób uruchomiania programów (2h).
 Edycja tekstu (2h).
 Praca w sieci: narzędzia ftp, telnet, mail (2h).
 Pierwszy program (2h).
 Środowisko uruchomieniowe (2h).
 Rozdanie tematów zadań i ich analiza; zakres tematyczny: podstawowe instrukcje sterujące i dekompozycja na funkcje (2h).
 Implementacja programów (2h).
 Testowanie i odbiór programów (2h).
 Rozdanie tematów zadań i ich analiza; zakres tematyczny: złożone algorytmy i struktury danych (2h).
 Weryfikacja specyfikacji wstępnej programu (2h).
 Dekompozycja na funkcje i struktury danych (2h).
 Implementacja funkcji (2h).
 Podział kodu źródłowego na pliki (2h).
 Uruchamianie i testowanie programów (2h).
 Odbiór programów i dokumentacji (2h).

**Metody oceny:**

2 kolokwia wykładowe oceniane w skali 0-25 każde
laboratoria oceniane w skali 0-50

suma ocen przekłada się na ocenę wg skali:
0-50 -> 2
51-60 -> 3
61-70 -> 3.5
71-80 -> 4
81-90 -> 4.5
91-100 -> 5

warunkiem koniecznym zaliczenia jest uzyskanie po minimum 13 punktów z laboratorium i z wykładu

studenci mają możliwość przystąpić do kolokwium poprawkowego, ocenianego w skali 0-100. Ocena z tego kolokwium jest podstawą oceny z przedmiotu, niezależnie od punktów uzyskanych w czasie semestru.

kolokwia sa pisemne z notatkami.

**Egzamin:**

nie

**Literatura:**

 Wirth, N.: Algorytmy + struktury danych = programy.
 Kernighan, B., Ritche, D.: Język ANSI C.
 Silvester, P.: System operacyjny unix.

**Witryna www przedmiotu:**

zmienia się wraz z prowadzącym

**Uwagi:**

przedmiot prowadzony na spółkę przez IMiO, IRE i ISE

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt Wpisz opis:**

znajomość języka C

Weryfikacja:

kolokwia i laboratorium

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**

### Profil ogólnoakademicki - umiejętności

**Efekt Wpisz opis:**

umięjętność pisania i uruchamiania programów komputerowych

Weryfikacja:

laboratorium

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt Wpisz opis:**

Wpisz opis

Weryfikacja:

Wpisz opis

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**