**Nazwa przedmiotu:**

Podstawy oprogramowania

**Koordynator przedmiotu:**

Robert KURJATA

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Inżynieria Biomedyczna

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

PPR

**Semestr nominalny:**

1 / rok ak. 2018/2019

**Liczba punktów ECTS:**

2

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

obecność na wykładach 30 godz.,
obecność na laboratorium 30 godz,
konsultacje przed kolokwiami i sprawdzianem praktycznym 5 godz.
praca własna studenta – 25 godz., w tym
przygotowanie do laboratoriów 15 godz.,
przygotowanie do sprawdzianów 10 godz.
Łączny nakład pracy studenta wynosi 90 godz., co odpowiada 3 pkt. ECTS.

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

Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli
akademickich: 2,6 pkt. ECTS, co odpowiada 65 godz. kontaktowym.

**Język prowadzenia zajęć:**

polski

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

1,8 pkt. ECTS, co odpowiada 45 godz. (laboratorium 30h, przygotowanie do laboratorium 15h)

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

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

**Wymagania wstępne:**

brak

**Limit liczby studentów:**

90

**Cel przedmiotu:**

Celem przedmiotu jest zapoznanie studentów z podstawami programowania strukturalnego i algorytmiki. Dodatkowo w ramach przedmiotu przedstawiane są podstawy systemu operacyjnego UNIX (na przykładzie systemu GNU/Linux) niezbędne do pracy w tym systemie w ramach laboratoriów. W zakresie podstaw programowania oczekuje się opanowania przez studentów procesu przejścia z opisu rzeczywistego zadania przez abstrakcyjny opis algorytmu na jego zapis formalny w języku C. Student posiądzie wiedzę i umiejętności w zakresie podstaw tworzenia algorytmów oraz praktycznego wykorzystania języka programowania na wybranych, elementarnych przykładach. Wybór języka C podyktowany jest tym, że stanowi on podstawę wielu języków wyższego poziomu, a także jest szeroko wykorzystywany w technice mikroprocesorowej. Bazowym standardem dla wykładu jest wersja C99.

**Treści kształcenia:**

 Treść wykładu:
1. Wprowadzenie do systemu UNIX (4h)
a. Zapoznanie się ze środowiskiem UNIX - procesy, uruchamianie i zatrzymywanie
programów, separacja użytkowników, organizacja systemu plików.
b. Podstawowe polecenia konsoli w zakresie obsługi plików, obsługi potoków i
komunikacji sieciowej.
2. Programowanie strukturalne w języku C (24h)
a. Podstawowe pojęcia: algorytm, program. Przekładanie opisu rzeczywistości na
algorytm komputerowy. Określenie złożoności obliczeniowej algorytmów na
wybranych przykładach. Algorytmy iteracyjne i rekurencyjne.
b. Proces przetwarzania tekstu programu na postać zrozumiałą dla maszyny -
preprocessing, kompilacja i konsolidacja. Zarządzanie kodem programu.
c. Podstawy leksyki i gramatyki języka C. Stałe: typ i wartość. Zmienne: typ, wartość,
nazwa. Typy podstawowe języka C. Niejawne rzutowanie typów. Pojęcie deklaracji i
definicji zmiennej (klasa extern). Wyrażenia i instrukcje. Priorytety i łączność
operatorów. Operatory arytmetyczne i przyrównania. Operator podstawienia. Lwartości.
Operatory post- i pre-inkrementacji, dekrementacji. Operatory bitowe i logiczne.
d. Blok kodu, funkcja. Zmienna lokalna i globalna. Zasięg i "żywotność" zmiennych.
Klasy zmiennych: static, auto, register. Instrukcja języka C - warunkowa if/else,
switch. Instrukcja break. Instrukcje iteracyjne: for, while, do/while. Instrukcja
continue.
e. Tablica. Struktura, unia. Instrukcja typedef. Pojęcie wskaźnika. Tablice i wskaźniki.
"Typ" łańcuchowy. Operatory referencji i dereferencji. Arytmetyka wskaźników.
Rzutowanie wskaźników. Przekazywanie argumentu funkcji przez wskazanie.
Przekazywanie tablicy do funkcji. Dynamiczna alokacja pamięci. Funkcje malloc i
free. Wejście i wyjście w C. Obsługa plików. Synchronizacja zawartości strumienia i
zawartości pliku.
f. Algorytmy i struktury danych: lista dowiązaniowa, sortowanie, drzewo, algorytmy
matematyczne.
3. Sprawdziany (2h)
Laboratorium
1. Wprowadzenie do systemu UNIX. (4h)
2. System plików, procesy, sposób uruchamiania programów. Edycja tekstu. (2h)
3. Pierwszy program. Środowisko uruchomieniowe. Podział kodu na fragmenty. Zarządzanie
kompilacją i konsolidacją. Narzędzie make. (2h)
4. Implementacja prostych zadań programistycznych. (2h)
5. Debugger - uruchamianie, wyszukiwanie błędów i analiza działania programów. (4h)
6. Implementacja prostych zadań programistycznych (4h)
7. Struktury, implementacja listy dowiązaniowej, dynamiczne zarządzanie pamięcią. (4h)
8. Implementacja wybranego algorytmu (np. sortowania, drzewa poszukiwań, struktury
słownikowej). (4h)
9. Sprawdzian praktyczny (2h)

**Metody oceny:**

kolokwia, laboratoria , sprawdzian praktyczny

**Egzamin:**

nie

**Literatura:**

● B.W. Kernighan, D.M. Ritchie, Język ANSI C Wydanie 2, Helion, Warszawa, 2010.
● Clovis L. Tondo, Scott E. Gimpel, Język ANSI C. Programowanie. Ćwiczenia. Wydanie II,
Helion, 2010
● Stephen Prata, Język C. Szkoła programowania. Wydanie VI, Helion, 2016,
● Richard Reese, Wskaźniki w języku C. Przewodnik, Helion, 2014,
● Robert Sedgewick, Kevin Wayne, Algorytmy. Wydanie IV, Helion, 2012

**Witryna www przedmiotu:**

nie ma

**Uwagi:**

brak

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka w01:**

Posiada podstawową wiedzę w zakresie użytkowania systemu operacyjnego UNIX i podstaw informatyki

Weryfikacja:

..

**Powiązane charakterystyki kierunkowe:** K\_W01, K\_W04

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

**Charakterystyka W02:**

Posiada podstawową wiedzę w zakresie programowania strukturalnego

Weryfikacja:

..

**Powiązane charakterystyki kierunkowe:** K\_W04

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

**Charakterystyka w03:**

Posiada podstawową wiedzę w zakresie konstruowania algorytmów komputerowych i ich złożoności

Weryfikacja:

..

**Powiązane charakterystyki kierunkowe:** K\_W04

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

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U01:**

Potrafi formułować zagadnienia w postaci algorytmicznej i zapisywać algorytmy w językach
programowania

Weryfikacja:

..

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U05

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, I.P6S\_UU

**Charakterystyka U02:**

Potrafi samodzielnie zdobywać informacje z dostępnych źródeł, ma umiejętność samokształcenia

Weryfikacja:

..

**Powiązane charakterystyki kierunkowe:** K\_U05

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

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka K01:**

Potrafi organizować własną pracę i pracować w małej grupie

Weryfikacja:

..

**Powiązane charakterystyki kierunkowe:** K\_K07

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

**Charakterystyka k02:**

Rozumie potrzebę dokształcania się przez całe życie

Weryfikacja:

..

**Powiązane charakterystyki kierunkowe:**

**Powiązane charakterystyki obszarowe:**