**Nazwa przedmiotu:**

Języki programowania

**Koordynator przedmiotu:**

Dr hab. inż. Waldemar SMOLIK

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Inżynieria Biomedyczna

**Grupa przedmiotów:**

Obowiązkowe

**Kod przedmiotu:**

JP

**Semestr nominalny:**

2 / rok ak. 2017/2018

**Liczba punktów ECTS:**

5

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

1) Liczba godzin bezpośrednich - 75 godz., w tym:
• wykład 30 godz.,
• laboratorium: 15 godz.,
• projekt: 15 godz.
• konsultacje – 10 godz.
• obrona projektów – 5 godz.
2) Praca własna - 50 godz.
• przygotowanie do wykładu: 10 godz.,
• przygotowanie projektu: 15 godz.,
• przygotowanie i sprawozdania do laboratoriów: 15godz.;
• przygotowanie do kolokwiów: 10 godz.
Razem 125 godz. (5 ECTS)

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

3 punkty ECTS - -75, w tym:
• wykład 30 godz,
• laboratorium: 15 godz.,
• projekt: 15 godz.
• konsultacje – 10 godz.
• obrona projektów – 5 godz.

**Język prowadzenia zajęć:**

polski

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

2,5 punkty ECTS - 55 godz.– w tym:
• wykonanie ćwiczeń w laboratorium: 15 godz.
• zajęcia projetowe: 15 godz.
• przygotowanie projektów: 15 godz.
• obrona projektów: 5 godz.
• przygotowanie i sprawozdania do laboratoriów: 15 godz.

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

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

**Wymagania wstępne:**

brak

**Limit liczby studentów:**

brak

**Cel przedmiotu:**

Celem przedmiotu jest przekazanie słuchaczom zasad konstruowania programów w językach strukturalnych
i obiektowych. Języki strukturalne zostaną omówione na przykładzie języka C, a języki obiektowe na przykładzie C++. W ramach laboratorium studenci nabędą umiejętność tworzenia programów narzędziowych z graficznym interfejsem użytkownika.

**Treści kształcenia:**

Zakres wykładu 1. Podstawowe pojęcia: algorytm, program. Stałe: typ i wartość. Zmienne: typ, wartość, nazwa. Typy podstawowe języka C. Niejawne rzutowanie typów. Pojecie deklaracji i definicji zmiennej. Wyrażenia
i instrukcje. Priorytety i łączność operatorów. Operatory arytmetyczne i przyrównania. Operator podstawienia. L-wartości. Operatory post- i preinkrementacji, dekrementacji. Operatory bitowe i logiczne. 2. Blok kodu, funkcja. Funkcja main z argumentami wywołania. Zmienna lokalna i globalna. Zasięg i "żywotność" zmiennych. Klasy zmiennych: static, auto, register, zmienne globalne. Deklaracja zmiennej jako extern. Instrukcja warunkowa if/else, switch. Instrukcja break. Instrukcje iteracyjne: for, while, do/while. Instrukcja continue. 3. Tablica. Struktura, unia. Instrukcja typedef. Pojęcie wskaźnika. Tablice a 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. Lista dowiązaniowa dwukierunkowa. 4. Wejście i wyjście w C. Obsługa plików. Tryby otwarcia: tekstowy i binarny. Synchronizacja zawartości strumienia i zawartości pliku. Preprocesor. 5. Sprawdzian 1 (20 pkt) 6. Podstawowe pojęcia: klasa, obiekt, kapsułkowanie, dziedziczenie. Paradygmat projektowania obiektowego. Zasady dostępu w klasie: pola prywatne i publiczne. Kontekst wprowadzany przez klasę. Konstruktory i destruktory. Konstruktor kopiujący. 7. Zasadnicze rozszerzenia w stosunku do C. Referencja i wskaźnik. Modyfikator const. Identyfikacja funkcji przez nagłówek. Przeciążanie funkcji. Przeciążanie operatorów. Domyślne wartości parametrów formalnych. 8. Strumieniowe wejście/wyjście. Funkcje i klasy zaprzyjaźnione. Wyjątki: sposób zgłaszania i przechwytywania, dziedziczenie wyjątków. 9. Dziedziczenie. Dziedziczenie wielokrotne klasy bazowej. Zasady dostępu do pól klasy bazowej. Sposób realizacji dziedziczenia. Wirtualne dziedziczenie. Wirtualne metody. Szablony. Standardowa biblioteka klas szablonowych. 10. Graficzny interfejs użytkownika. Systemy „okienkowe”. Obsługa urządzeń graficznych na przykładzie wybranej biblioteki. 11. Sprawdzian 2 (20 pkt)

**Metody oceny:**

Laboratorium - 60 pkt Sprawdzian 1 - 20 pkt
Sprawdzian 2 - 20 pkt Punkty są przeliczne na
oceny wg tabeli 0-50 -> 2 51-60 -> 3 61-70 -> 3.5
71-80 -> 4 81-90 -> 4.5 91-100 -> 5

**Egzamin:**

nie

**Literatura:**

1. B. Kernighan, D. Ritchie, Język ANSI C, WNT, 2004
2. B. Stroustrup, Język C++, WNT, 2002
3. N. Wirth, Algorytmy + struktury danych = programy
4. P. Wróblewski, Algorytmy. Struktury danych i techniki programowania, Helion 2003
5. A. Lippmann, Programowanie obiektowe
6. P. Silvester, System operacyjny unix.
7. S. Lippman, Podstawy języka C++, WNT, 2003
8. S. Prata,Język C++. Szkoła programowania, Wydanie V, Helion, 2006
9. H. Schild, C++, ReadMe, 2002
10. N. Josuttis, C++. Biblioteka standardowa. Podręcznik programisty; Helion, 2003
11. S. Mayers, STL w praktyce. 50 sposobów efektywnego wykorzystania; Helion, 2004
12. R. Lischner, STL. Leksykon kieszonkowy, Helion,
O’Reilly, 2004

**Witryna www przedmiotu:**

www.ire.pw.edu.pl/~wsmolik/JezykiProgramowania

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt JP\_W01:**

Znajomość języka C i C++

Weryfikacja:

dwa sprawdziany, laboratorium, projekt

**Powiązane efekty kierunkowe:** K\_W04

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

### Profil ogólnoakademicki - umiejętności

**Efekt JP\_U1:**

Umiejętność projektowania i kodowania programów
w języku strukturalnym C i obiektowym C++

Weryfikacja:

laboratorium, projekt

**Powiązane efekty kierunkowe:** K\_U02, K\_U03, K\_U06

**Powiązane efekty obszarowe:** T1A\_U02, T1A\_U07, T1A\_U03, T1A\_U06, T1A\_U07, T1A\_U09