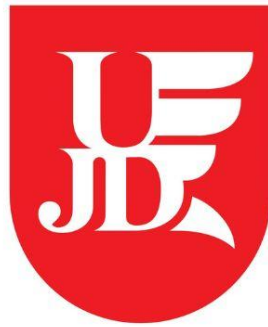


UNIwersytet JANA DŁUGOSZA w CZĘSTOCHOWIE



Wydział Nauk Ścisłych, Przyrodniczych i Technicznych

Kierunek: Informatyka
Specjalność: Inżynieria Oprogramowania

Zuzanna Hynek

Laboratorium 2

Programowanie w C i C++

Częstochowa, 2025

Zadanie 1

Napisz program wyświetlający dwa napisy "Witaj świecie" oraz "Programowanie w C/C++" każdy w osobnej linii. Użyj do tego jednej instrukcji do wyświetlania ciągów. Kolejno wyświetl trzy dowolne linie tekstu a po każdej z nich informację "Naciśnij Enter aby kontynuować...". Po tym napisie program ma oczekiwać na naciśnięcie przez użytkownika dowolnego klawisza. Zadbaj o czyszczenie bufora klawiatury. Można użyć funkcji `getchar()` [stdio] a następnie `getche()` [conio] i zaobserwować różnicę podczas naciśnięcia klawisza np litery i samego klawisza enter.

```
#include <stdio.h>
#include <conio.h>

int main() {
    printf("Witaj świecie\nProgramowanie w C/C++\n");

    printf("Pierwsza linia tekstu.\n");
    printf("Naciśnij Enter aby kontynuować...\n");
    getchar();

    printf("Druga linia tekstu.\n");
    printf("Naciśnij Enter aby kontynuować...\n");
    getchar();

    printf("Trzecia linia tekstu.\n");
    printf("Naciśnij Enter aby kontynuować...\n");
    getchar();
    char c = getche();
    printf("\nNaciśnąłeś klawisz: %c\n", c);

    return 0;
}
```

Zadanie 2

Stworzyć nowy projekt, nazwać go Projekt1 i i zapisać. Wkleić poniższy kod

```
#include <iostream>
#include <stdio.h>
```

```
using namespace std;int i,j;int k;int main(){for (j=1; j<=10; j++){for
(i=1;i<=10;i++){k=i*j;printf("%4d",k);}cout<<endl;}return 0;}
```

Przeprowadzić formatowanie ręczne zgodnie z dobrymi praktykami programowania tak aby każdy wiersz zaczynał się w kolumnie zero. Następnie wykonać autoformatowanie za pomocą wtyczki *AStyle*. Skompilować i sprawdzić co podaje kompilator w oknie statusu *Logs & others*. Sprawdzić wynik działania programu.

```
#include <iostream>
#include <stdio.h>

using namespace std;
int i,j;
int k;

int main(){
    for (j=1; j<=10; j++){
        for (i=1;i<=10;i++){
            k=i*j;
            printf("%4d", k);
        }
        cout<<endl;
    }
    return 0;
}
```

Zadanie 3

Stworzyć nowy projekt i wkleić poniższy kod:

```
#include <iostream>
#include <stdio.h>
```

```
using namespace std;
```

```
int i,j;
float K;
```

```
int main()
{
    for (j=1; j<=10; j++)
    {
        for (i=1; i<=10; i++)
        {
            k=i*j
            printf("%4d",k);
        }
        cout << endl;
    }
    //return 0;
}
```

Przeprowadzić proces kompilacji. Jeśli pojawią się błędy opisać je i poprawić.

```
#include <iostream>
```

```
using namespace std;
```

```
int i, j, k; // zmienna k jako int
```

```
int main()
{
    for (j = 1; j <= 10; j++) {
    for (i = 1; i <= 10; i++) { // brakowało nawiasu
    k = i * j; // brakowało średnika
    printf("%4d", k);
    }
    cout << endl;
    }
    return 0;
}
```

Zadanie 4

Napisz program dzielący dwie dowolne liczby całkowite przez siebie, wynik ma być liczbą rzeczywistą. Skorzystaj ze stałych lub literałów liczbowych. Przeanalizuj wyniki i sposób działania operacji dzielenia w C++.

```
#include <iostream>

using namespace std;

int main() {
    int a = 10;
    int b = 4;

    float result;

    result = static_cast<float>(a) / b;

    cout << "Wynik dzielenia to: " << result << endl;

    return 0;
}
```

Zadanie 5

Napisać program wyświetlający ile bajtów zajmują dostępne w Twoim kompilatorze i wersji języka typy danych. Sprawdź typy: bool, char, unsigned char, int, unsigned int, long int, long long int, unsigned long long int, float, double, long double.

```
#include <iostream>

using namespace std;

int main() {
    cout << "Rozmiar typu bool: " << sizeof(bool) << " bajtów" << endl;
    cout << "Rozmiar typu char: " << sizeof(char) << " bajtów" << endl;
    cout << "Rozmiar typu unsigned char: " << sizeof(unsigned char) << " bajtów" << endl;
    cout << "Rozmiar typu int: " << sizeof(int) << " bajtów" << endl;
    cout << "Rozmiar typu unsigned int: " << sizeof(unsigned int) << " bajtów" << endl;
    cout << "Rozmiar typu long int: " << sizeof(long int) << " bajtów" << endl;
    cout << "Rozmiar typu long long int: " << sizeof(long long int) << " bajtów" << endl;
    cout << "Rozmiar typu unsigned long long int: " << sizeof(unsigned long long int) << " bajtów" << endl;
    cout << "Rozmiar typu float: " << sizeof(float) << " bajtów" << endl;
    cout << "Rozmiar typu double: " << sizeof(double) << " bajtów" << endl;
    cout << "Rozmiar typu long double: " << sizeof(long double) << " bajtów" << endl;

    return 0;
}
```

Zadanie 6

Napisać program sprawdzający czy podana wartość znajduje się w przedziale:

<-10..20)

(0..10>

<-3..3>

Podana wartość może występować w programie jako stała lub zmienna z konkretną wartością. Sprawdzić wynik dla wartości wewnętrznych i granicznych: -10,20,0,10,-3,3. Odpowiedzią może być 0 lub 1.

```
#include <iostream>

using namespace std;

bool jeden(int value) {
    return (value > -10 && value < 20);
}

bool dwa(int value) {
    return (value > 0 && value <= 10);
}

bool trzy(int value) {
    return (value >= -3 && value <= 3);
}

int main() {
    int values[] = {-10, 20, 0, 10, -3, 3};

    for (int i = 0; i < 6; i++) {
        int value = values[i];
        cout << "Wartość: " << value << endl;
        cout << "Przedział 1: " << (jeden(value) ? 1 : 0) << endl;
        cout << "Przedział 2: " << (dwa(value) ? 1 : 0) << endl;
        cout << "Przedział 3: " << (trzy(value) ? 1 : 0) << endl;
        cout << endl;
    }

    return 0;
}
```

Zadanie 7

Sprawdzić czy podana wartość występuje w którymś z przedziałów:

$[-10..22) \cup (30..40]$

$(0..3) \cup (5..7]$

Sprawdzić wynik dla wartości wewnętrznych i granicznych podanych przedziałów. Odpowiedzią może być 0 lub 1. Nie używamy instrukcji warunkowej.

```
#include <iostream>

using namespace std;

int main() {
    int x;
    cout << "Podaj liczbę: ";
    cin >> x;

    int wynik = (-10 <= x && x < 22) || (30 < x && x <= 40) || (0 < x && x <= 3) || (5 <= x &&
x < 7);

    cout << "Wynik: " << wynik << endl;

    return 0;
}
```

Zadanie 8

Napisz program obliczający sumę i średnią trzech liczb (literały, stałe). Uzasadnij dobór typów danych argumentów i wyniku. Rozbuduj program o operacje odejmowania na stałych i zmiennych typów int oraz unsigned int.

Przeprowadź test dla odejmowania liczby większej od mniejszej zwłaszcza dla typów unsigned oraz mieszanych signed i unsigned. Skorzystaj ze stałych definiowanych za pomocą notacji C (stałe symboliczne, dyrektywy preprocesora) oraz C++ a także modyfikatorów typów tych stałych.

```
#include <iostream>

using namespace std;

#define liczba1 15
const int liczba2 = 25;
constexpr int liczba3 = 35;

int main() {
    int suma = liczba1 + liczba2 + liczba3;
    double srednia = suma / 3.0;

    cout << suma << endl;
    cout << srednia << endl;

    const int a = 50;
    const unsigned int b = 25, c = 100;

    cout << a - b << endl;
    cout << b - c << endl;
    cout << a - c << endl;

    cin.get();
    return 0;
}
```

Zadanie 9

Napisz program obliczający tygodniowe całkowite wynagrodzenie. Uwzględnij godziny i stawki standardowe oraz nadliczbowe. Zadeklaruj zmienne, stałe oraz dobrać ich typy danych. Program ma podać ile pracownik zarobił w godzinach podstawowych, ile w nadliczbowych, ile w sumie oraz jaki procent godzin standardowych stanowią godziny nadliczbowe. Zakładamy, że stawka nadliczbową jest o 50% większa od standardowej.

```
#include <iostream>
using namespace std;

#define STAWKA 20.0
const int GODZINY = 40;
constexpr double NADLICZBOWE = 1.5;

int main() {
    double std, nad;

    cout << "Podaj godziny standardowe: ";
    cin >> std;

    cout << "Podaj godziny nadliczbowe: ";
    cin >> nad;

    double wyn_std = std * STAWKA;
    double wyn_nad = nad * STAWKA * NADLICZBOWE;
    double suma = wyn_std + wyn_nad;

    cout << "\nStandardowe: " << wyn_std << " PLN\n";
    cout << "Nadliczbowe: " << wyn_nad << " PLN\n";
    cout << "Suma: " << suma << " PLN\n";
    cout << "Procent nadliczbowych: " << (nad / std) * 100 << " %\n";

    return 0;
}
```

Zadanie 10

Napisać program wyświetlający znak 'A' za pomocą zmiennej typu int oraz wartość 97 za pomocą zmiennej typu char.

```
#include <iostream>
using namespace std;

int main() {
    int a = 65;
    char b = 97;

    cout << (char)a << endl;
    cout << b << endl;

    return 0;
}
```

Zadanie 11

Zdefiniuj typy zmiennych dla następujących wartości:

- 3999999999
- 'a'
- 23
- 0.3
- 999e999

Napisz program wyświetlający wartości takich zmiennych.

```
#include <iostream>

using namespace std;

int main() {
    long long int a = 3999999999;
    char b = 'a';
    int c = 23;
    double d = 0.3;
    long double e = 999e999;

    cout << "3999999999: " << a << endl;
    cout << "'a': " << b << endl;
    cout << "23: " << c << endl;
    cout << "0.3: " << d << endl;
    cout << "999e999: " << e << endl;

    return 0;
}
```

Zadanie 12

Napisz program przedstawiający efekty wyjścia poza zakres zmiennych typów w stronę -inf jak i +inf (użyj też modyfikatora unsigned):

- int
- bool (0-fałsz, wszystko inne prawda)
- long
- char
- float

Wykorzystaj stałe takie jak INT_MAX, INT_MIN dostępne w Twoim kompilatorze (np biblioteka limits.h, float.h) lub sprawdź samodzielnie dostępne zakresy.

Do zmiennej typu int wpisz wartość maksymalną, dodaj jeden, wyświetl wartość tej zmiennej. Efekt? Wnioski?

```
#include <iostream>
#include <climits>
#include <float.h>
using namespace std;

int main() {
    int i = INT_MAX; i++; cout << "int: " << i << endl;
    bool b = 100; cout << "bool: " << b << endl;
    long l = LONG_MAX; l++; cout << "long: " << l << endl;
    char c = CHAR_MAX; c++; cout << "char: " << int(c) << endl;
    float f = FLT_MAX; f *= 2; cout << "float: " << f << endl;
    unsigned u = UINT_MAX; u++; cout << "unsigned: " << u << endl;
    return 0;
}
```


Zadanie 12a

Napisz program pokazujący efekty użycia operatora unarnego pre i post inkrementacji użytego w wyrażeniu. Przykład: zmienna a ma wartość 10. Napisz kod (jedna instrukcja) wyświetlający wartość tej zmiennej oraz zmiennej potraktowanej operatorem preinkrementacji np: `cout << a << " " << ++a << endl;`. Wnioski? Jak zatem wygląda kolejność przetwarzania zmiennych i wykonywania instrukcji.

```
#include <iostream>
using namespace std;

int main() {
    int a = 10;
    cout << a << " " << ++a << " " << a++ << " " << a << endl;
    return 0;
}
```

Zadanie 13

Napisać program obliczający kwotę netto z podanej przez użytkownika kwoty brutto. Wartość podatku wyrażoną procentowo podaje także użytkownik. (zmienne rzeczywiste) Wynikiem ma być informacja typu: "Przy podatku 23% z kwoty brutto 123PLN otrzymujemy kwotę netto równą 100PLN. Wartość podatku to 23PLN". Wartości mają być wyświetlane z dokładnością do drugiego miejsca po przecinku. Zadbaj o odpowiednie komunikaty.

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    double brutto, podatek, netto, podatekWartosc;

    cout << "Kwota brutto: ";
    cin >> brutto;
    cout << "Procent podatku: ";
    cin >> podatek;

    podatekWartosc = brutto * podatek / 100;
    netto = brutto - podatekWartosc;

    cout << fixed << setprecision(2);
    cout << "Przy podatku " << podatek << "% z kwoty brutto " << brutto
        << " PLN otrzymujemy kwotę netto " << netto
        << " PLN. Wartość podatku to " << podatekWartosc << " PLN." << endl;

    return 0;
}
```

Zadanie 14

Napisz program wczytujący z klawiatury liczbę rzeczywistą i wyświetlający jej wartość bezwzględną. Użyj funkcji standardowej umieszczonej w zewnętrznej bibliotece swojego środowiska IDE.

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    double liczba;

    cout << "Podaj liczbę: ";
    cin >> liczba;

    *
    cout << "Wartość bezwzględna liczby: " << abs(liczba) << endl;

    return 0;
}
```

Zadanie 15

Napisz program, który wczytuje ze standardowego wejścia liczbę całkowitą i wypisuje na standardowym wyjściu liczbę o jeden większą. Zastosować i omówić trzy sposoby inkrementacji w C++. Przedyskutować sposoby inkrementowania przedrostkowego i przyrostkowego w wyrażeniach.

```
#include <iostream>
using namespace std;

int main() {
    int liczba;

    cout << "Podaj liczbę całkowitą: ";
    cin >> liczba;

    cout << "Przyrostkowa: " << liczba++ << endl;
    cout << "Przedrostkowa: " << ++liczba << endl;
    cout << "Zwykłe dodanie: " << liczba + 1 << endl;

    return 0;
}
```

Zadanie 16

Napisać program przeliczający odległość podaną w km na mile. Zakładamy, że 1 mila to 1.6 km.

```
#include <iostream>
using namespace std;

int main() {
    double km, mile;

    cout << "Podaj odleglosc w km: ";
    cin >> km;

    mile = km / 1.6;
    cout << km << " km to " << mile << " mil" << endl;

    return 0;
}
```

Zadanie 17

Napisać program przyjmujący dane: długość trasy w km oraz ilość spalonego paliwa. Program jako wynik ma podać średnie zużycie paliwa na 100km. Zadbać o odpowiednie informacje dla użytkownika.

```
#include <iostream>
using namespace std;

int main() {
    double dlugoscTrasy, spalonePaliwo, srednieZuzycie;

    cout << "Podaj długość trasy w kilometrach: ";
    cin >> dlugoscTrasy;

    cout << "Podaj ilość spalonego paliwa w litrach: ";
    cin >> spalonePaliwo;

    srednieZuzycie = (spalonePaliwo / dlugoscTrasy) * 100;

    cout << "Średnie zużycie paliwa na 100 km: " << srednieZuzycie << " litrów." << endl;

    return 0;
}
```

Zadanie 19

Napisz program będący symulatorem operacji logicznej XOR. Na tym etapie nie dysponujesz operatorami and, or, xor, not ani instrukcją warunkową.

```
#include <iostream>
using namespace std;

int main() {
    int a, b;

    cout << "Podaj pierwsza liczbe (0 lub 1): ";
    cin >> a;
    cout << "Podaj druga liczbe (0 lub 1): ";
    cin >> b;

    int wynik = (a + b) % 2;
    cout << "XOR: " << wynik << endl;

    return 0;
}
```

Zadanie 21

Odpowiedz na pytania:

a) Czym się różni znak '3' od cyfry 3 w sensie języka C++

Znak '3' to pojedynczy znak o wartości ASCII 51, a cyfra 3 to cyfra o wartości 3.

b) Co to jest typ danych char i do czego go używamy

Char reprezentuje pojedynczy znak i jest przechowywany jako liczba całkowita odpowiadająca wartości ASCII tego znaku. Używamy go do przechowywania pojedynczych znaków w tekstach lub do operacji na znakach.

c) Co to jest typ int i jakie ma zastosowanie

Int przechowuje liczby całkowite i służy do obliczeń oraz przechowywania liczb bez części dziesiętnej.

d) Podaj różnicę między typem int a float

Int przechowuje liczby całkowite, a float liczby zmiennoprzecinkowe.

e) Jaki typ danych zastosujesz w przypadku wyznaczenia kwoty w PLN

Float.

f) Jakie zastosowanie ma stała a jakie zmienna

Stała ma wartość, która nie może się zmienić po jej przypisaniu, a zmienna może mieć zmienną wartość w trakcie działania programu.

g) Jak można zadeklarować stałą

Const.

h) Jak można zmienić wartość stałej

Nie można zmienić wartości stałej po jej zadeklarowaniu.

i) Podaj różnicę pomiędzy == a =

== to operator porównania, który sprawdza, czy dwie wartości są równe.

= to operator przypisania, który przypisuje wartość zmiennej.

j) Czym się różni typ od zmiennej

Typ to rodzaj danych, a zmienna to miejsce, gdzie te dane są przechowywane.

k) Omów działanie znanych modyfikatorów zmiennych

const – zmienna, której nie można zmienić.

static – zmienna zachowuje wartość między wywołaniami funkcji.

l) Które zmienne, ze znakiem czy bez znaku mogą przechowywać większą ilość danych? Podaj powody.
Zmienne bez znaku mogą przechowywać większą ilość danych, ponieważ wszystkie bity są przeznaczone na wartości dodatnie, a nie na znak.

m) Co to jest rzutowanie i jak się je przeprowadza?

Rzutowanie to konwersja jednej zmiennej na inny typ danych. Przeprowadza się je za pomocą operatorów rzutowania, np. (typ) zmienna lub `static_cast<typ>(zmienna)`.