



# Programowanie Obiektowe i Inżynieria Oprogramowania

## LAB02: Podstawy C++, definiowanie prostych funkcji

Wiktor Jachym  
12.03.2025

# W dużym skrócie o C++



- C++ jest jednym z popularniejszych języków programowania
- Znajduje zastosowanie w wielu dziedzinach ze względu na swoją wydajność, kontrolę nad zasobami oraz wsparcie dla programowania obiektowego. Przykłady: oprogramowanie wbudowane i IoT, tworzenie gier, aplikacje naukowe i inżynieryjne (Matlab jest częściowo napisany w C++), przeglądarki internetowe, itd...
- C++ został opracowany jako rozszerzenie języka C (m. in. dodano OOP), a oba języki mają niemal identyczną składnię





- Login: user
- Hasło: student2000



# Pierwszy program w DEV C++ 6.3



Dołączenie biblioteki we/wy

Komentarz

```
#include <iostream>
/* run this program using the console pauser or add your own getch, system("pause") or input loop */
int main(int argc, char** argv) {
    std::cout << "Hello World!";
    return 0;
}
```

Zwraca wartość 0 jako  
wynik działania programu

Deklaracja funkcji main()

Zamyka funkcję main()

Obiekt z biblioteki *iostream* i przestrzeni nazw *std*, używany do wypisywania danych na standardowe wyjście (w tym wypadku konsolę). Używany wraz z operatorem <<, służącym do „wydrukowania” tekstu

# Mała uwaga



```
#include <iostream>

/* run this program using the console pauser or add your own getch, system("pause") or input loop */

int main(int argc, char** argv) {

    std::cout << "Hello World!";

    return 0;

}
```



```
#include <iostream>
int main(int argc, char** argv) { std::cout << "Hello World!"; return 0; }
```

## Ale jest mniej elegancko!

(Jest to możliwe, gdyż kompilator ignoruje puste wiersze)

# Krótko i szybko o zmiennych



```
#include <iostream>
#include <string>

using namespace std;

int main(int argc, char** argv) {

    int a;
    a = 11;

    // ale możemy też zadeklarować i przypisać wartość do zmiennej od razu:
    // int a = 5;

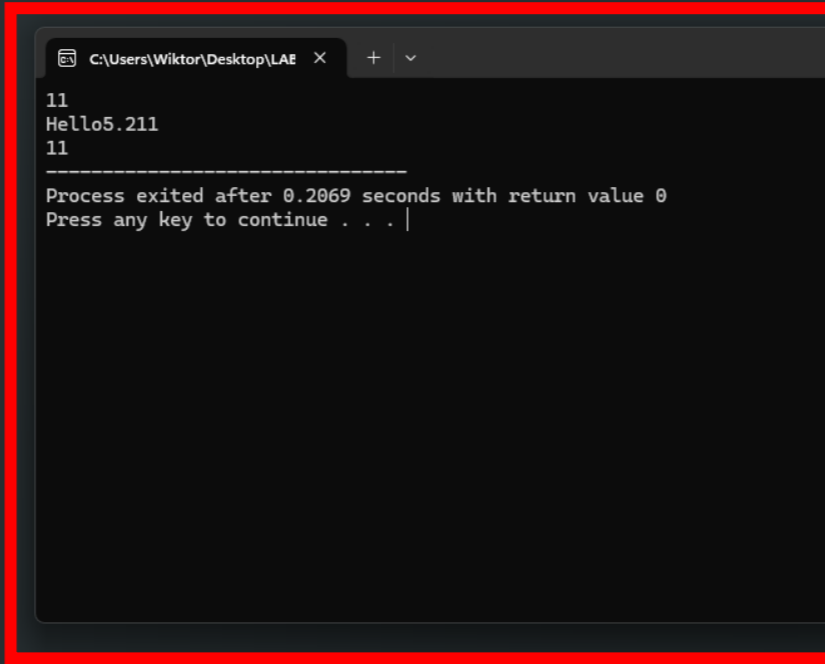
    double b = 5.2;
    char c = 'W';
    string d = "Hello";
    bool e = "true";

    cout << a << endl;

    cout << d << b << a << endl;

    int x = 5;
    int y = 6;
    int sum = x + y;
    cout << sum;

    return 0;
}
```



```
C:\Users\Wiktork\Desktop\LAE x + v
11
Hello5.211
11
-----
Process exited after 0.2069 seconds with return value 0
Press any key to continue . . . |
```

```
int x = 5, y = 6, z = 50;
cout << x + y + z;
```

```
int x, y, z;
x = y = z = 50;
cout << x + y + z;
```

```
const double liczba_pi = 3.14;
```

Dodatkowo:

- Nazwa zmiennej może zawierać litery, cyfry, symbol podkreślenia
- Musi zaczynać się od cyfry lub symbolu podkreślenia
- Nazwy zmiennych są wrażliwe na wielkość liter
- Nie można używać nazw już zarezerwowanych, np. *int*
- Zaleca się używanie nazw opisowych w celu stworzenia zrozumiałego kodu (czyli nie robimy tak, jak ja na przykładzie 😊)



# Krótko i szybko o instrukcjach warunkowych



```
#include <iostream>

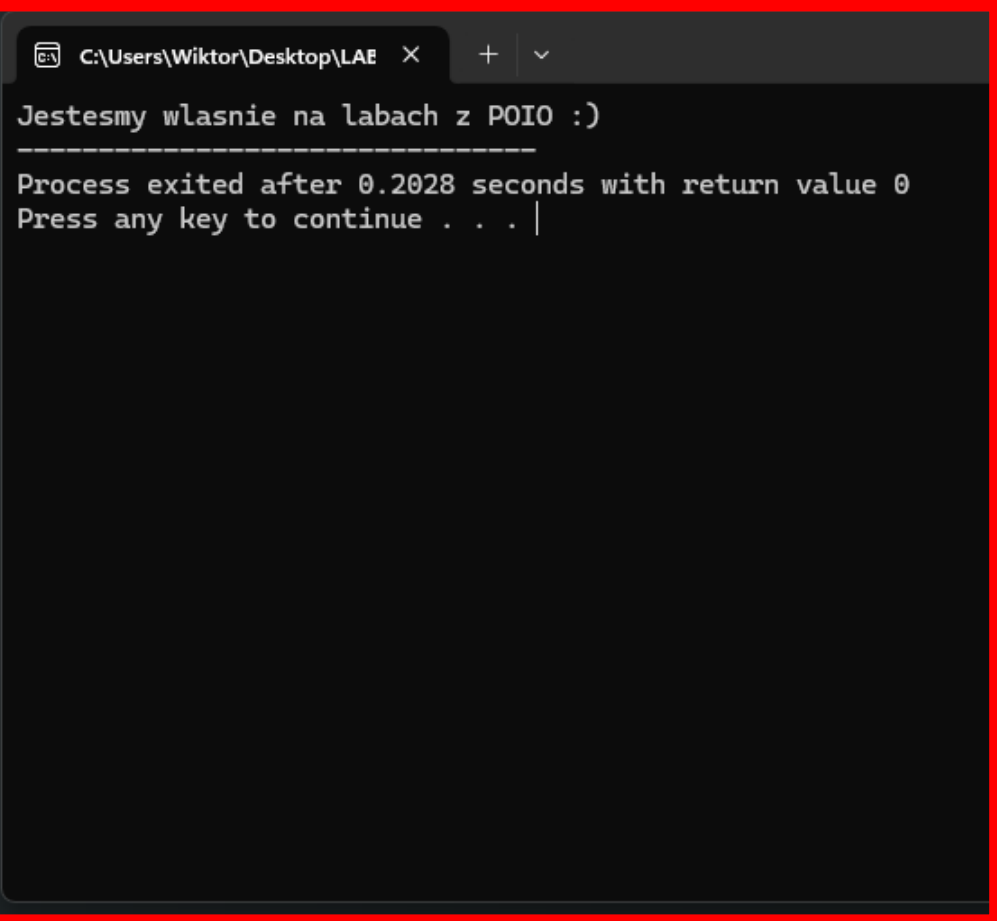
using namespace std;

int main(int argc, char** argv) {

    double godzina = 16.34;

    if (godzina < 15) {
        cout << "Nie jesteśmy jeszcze na labach z POIO";
    } else if (godzina >= 15 && godzina <= 17.15) {
        cout << "Jestemy właśnie na labach z POIO :)";
    } else if (godzina > 17.15) {
        cout << "Laby z POIO już się skończyły :(";
    }

    return 0;
}
```



```
C:\Users\Wiktork\Desktop\LAE x + v
Jestemy właśnie na labach z POIO :)
-----
Process exited after 0.2028 seconds with return value 0
Press any key to continue . . . |
```

# Krótko i szybko o pętlach



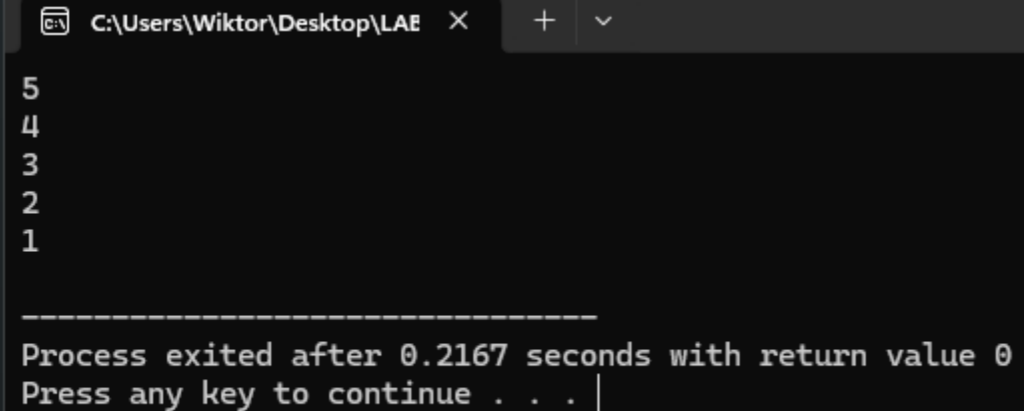
```
#include <iostream>

using namespace std;

int main(int argc, char** argv) {

    int i = 5;
    while (i > 0) {
        cout << i << endl;
        i--; // albo: i = i - 1;
    }

    return 0;
}
```



```
C:\Users\Wiktora\Desktop\LAE x + v
5
4
3
2
1
-----
Process exited after 0.2167 seconds with return value 0
Press any key to continue . . . |
```

# Funkcje



- Funkcja to blok kodu, który działa tylko wtedy, gdy zostanie wywołany
- Do funkcji można przekazywać dane zwane parametrami
- Funkcje służą do wykonywania określonych działań i **są ważne dla ponownego wykorzystania kodu**: Zdefiniuj kod raz i używaj go wiele razy

# Zadeklarujemy, zdefiniujemy i wywołamy prostą funkcję



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

void pierwszaFunkcja(int num_a, int num_b) {
    cout << num_a + num_b << endl;
}

int main(int argc, char** argv) {
    int a = 5;
    int b = 3;
    pierwszaFunkcja(a, b);
    return 0;
}
```

Deklaracja funkcji *pierwszaFunkcja*

void pierwszaFunkcja(int num\_a, int num\_b) {

cout << num\_a + num\_b << endl;

}

int main(int argc, char\*\* argv) {

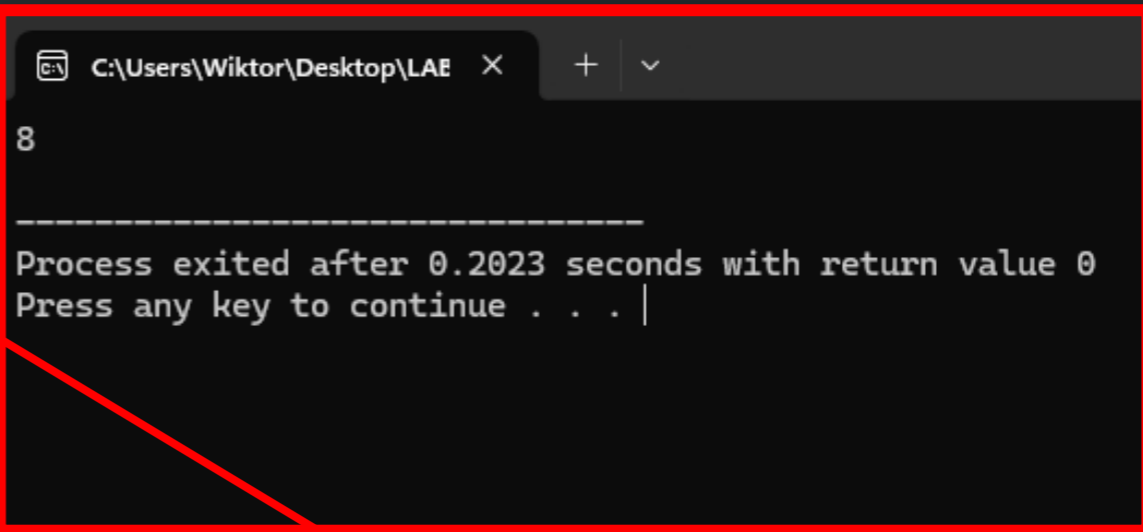
int a = 5;

int b = 3;

pierwszaFunkcja(a, b);

return 0;

}



Wywołanie naszej funkcji

Definicja funkcji *pierwszaFunkcja*



# Praca indywidualna



- Strona domowa: <https://home.agh.edu.pl/~wjachym/>
- Prezentacja: LAB02 -> LAB02\_intro.pdf
- Ćwiczenia: LAB02 -> LAB02\_ex.pdf
- Pomoc naukowa: <https://www.w3schools.com/cpp/>

**MIŁEJ PRACY! 😊**