



```
#include <iostream>
using namespace std;
...           // deklaracje stałych
...           // deklaracje funkcji
int main ()
{
    // część wykonawcza programu
    return 0;
}
```

Zadanie 1. Wyprowadzanie danych

Napisz program wyprowadzający na ekran napis: „jestem na konferencji”. Zapisz program w pliku pod nazwą *zadanie1*.

Wskazówki: Prosty program w języku C++ ma postać:

```
Zadanie1.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      cout << "Jestem na konferencji";
8      return 0;
9  }
```

W języku C++ `cout` jest obiektem reprezentującym standardowe wyjście.

Zadanie 2. Deklarowanie zmiennych, wprowadzanie danych, wykonywanie obliczeń i wyprowadzanie wyników na ekran

Napisz program umożliwiający wprowadzenie dwóch liczb całkowitych *a* i *b*, obliczenie ich sumy (*suma*) i wyprowadzenie wyniku (*suma*) na ekran monitora. Skompiluj i uruchom program. Zapisz program w pliku pod nazwą *zadanie2*.

Wskazówki:

W pokazanym fragmencie programu zadeklarowano trzy zmienne typu całkowitego:

```
Zadanie2.cpp
1  #include <iostream>
2
3  using namespace std;
4  int a,b,suma;
5  int main()
6  {
```

Przykład wprowadzania danych: `cin >> a >> b;`

W języku C++ `cin` jest obiektem reprezentującym standardowe wejście.

W instrukcji przypisania używamy znaku „=” np. `iloczyn=a*b;`

Warsztaty dla nauczycieli

Zadanie 3. Dodawanie opisów

Do programu z zadania 2. dodaj opisy „wprowadz pierwsza liczbe”, „wprowadz druga liczbe”, „suma wynosi”. Zapisz plik pod tą samą nazwą.

Wskazówki: Umieszczenie `endl` lub znaków `"\n"` na końcu polecenia `cout` zaczyna nowy wiersz, np.: `cout << "wprowadz pierwsza liczbe:" << "\n";`

Zadanie 4. Stosowanie instrukcji warunkowej

Napisz program, który umożliwi wprowadzenie z klawiatury dwóch liczb całkowitych (*liczba1*, *liczba2*) i wyprowadzenie na ekran większej z nich. Zapisz program w pliku pod nazwą *zadanie4*.

Wskazówki: Instrukcja warunkowa w języku C++ ma postać:

```
if (wyrażenie)
    instrukcja1;
else instrukcja2;
```

lub

```
if (wyrażenie) instrukcja1;
```

Na przykład:

```
13     if(liczba1>liczba2)
14         cout << "wieksza jest: " << liczba1;
15     else cout << "wieksza jest: " << liczba2;
16
```

Zadanie 5. Stosowanie instrukcji iteracyjnej for

Zmodyfikuj program z zadania 4, aby można było wprowadzić i sprawdzić kolejno dziesięć par liczb. Zapisz plik pod tą samą nazwą.

Wskazówki: Instrukcja iteracyjna `for` w języku C++ ma postać:

```
for (wyrażenie_początkowe; warunek; wyrażenie_pętli) instrukcja;
```

Fragment programu z zastosowaniem instrukcji `for`:

```
9     for (i=0;i<10;i++)
10    {
11        cout << "Podaj pierwsza liczbe: ";
12        cin >> liczba1;
13        cout << "Podaj druga liczbe: ";
14        cin >> liczba2;
15        if(liczba1>liczba2)
16            cout << "wieksza jest: " << liczba1 << "\n";
17        else cout << "wieksza jest: " << liczba2 << "\n";
18    }
```

Zadanie 6. Stosowanie instrukcji warunkowej if

Utwórz program sprawdzający, czy wprowadzona z klawiatury liczba jest dodatnia czy niedodatnia. Zadeklaruj zmienną typu rzeczywistego `float` o nazwie *liczba*. Jeśli wprowadzona z klawiatury wartość zmiennej *liczba* jest dodatnia, wyświetlaj na ekranie napis „dodatnia”, w przeciwnym wypadku – napis „niedodatnia”. Na koniec wyświetl na ekranie wartość zmiennej *liczba*. Zapisz program w pliku pod nazwą *Zadanie6*.

Warsztaty dla nauczycieli

Zadanie 7. Stosowanie instrukcji iteracyjnej `while`

Zmodyfikuj program z zadania 6, aby umożliwił wprowadzanie liczb dopóki wprowadzana liczba nie będzie równa zero. Jeśli wprowadzona z klawiatury wartość zmiennej *liczba* jest dodatnia, wyświetlaj na ekranie napis „dodatnia”, w przeciwnym wypadku – napis „ujemna”. Na koniec wyświetl na ekranie napis „zero”. Zapisz program w pliku pod nazwą *zadanie7*.

Wskazówki: Instrukcja iteracyjna `while` w języku C++ ma postać:

```
while (warunek) instrukcja
```

Najpierw sprawdzany jest *warunek*. Jeśli jest spełniony, to wykonywana jest *instrukcja*.

Na przykład:

```
cin >>a>>b;
while (b!=0)
{
    iloraz=a/b;
    cout << "Iloraz wynosi" << iloraz << endl;
    cin >> >> b;
}
```

Uwaga: Wewnątrz bloku *instrukcji* powinna być zawsze umieszczona instrukcja, która zmienia wartość *warunku* – w przeciwnym wypadku program się zapętli.

Zadanie 8. Stosowanie funkcji – definiowanie i wywoływanie

Napisz program, w którym zdefiniujesz funkcję *różnica_kwadrat* z dwoma parametrami *a* i *b*, obliczającą różnicę kwadratów dwóch liczb całkowitych. Wywołaj funkcję w programie głównym z parametrami aktualnymi *x* i *y*. Zapisz program w pliku pod nazwą *zadanie8*.

Wskazówki:

W języku C++ wszystkie podprogramy nazywane są **funkcjami**. Dzielimy je na dwie grupy: funkcje niezwracające wartości i funkcje zwracające wartość.

Funkcja zwracająca wartość wylicza wartość i odsyła tę wartość do funkcji wywołującej. Mówimy, że funkcja zwraca wartość. Funkcja zwracająca wartość musi zawierać instrukcję `return` z wartością, która ma być zwrócona do funkcji wywołującej. Zwracana wartość może być stałą, zmienną lub wyrażeniem. Na przykład:

```
return a*b;
return Wynik;
```

W języku C++ funkcję można wywołać, umieszczając wynik w zmiennej. Na przykład:

```
Wynik=RoznicaKwadrat();
```

Funkcję można wykorzystać jako element wyrażenia lub instrukcji. Na przykład:

```
cout << RoznicaKwadrat();
```

Uwaga: **Parametr** to wartość przekazywana funkcji, a **wartość zwracana** to wartość przekazywana z funkcji.

```
1  #include <iostream>
2  using namespace std;
3  int x, y;
4  int RoznicaKwadrat(int a, int b)
5  {
6      return x*x - y*y;
7  }
8
9  int main()
10 {
11     cin >> x >> y;
12     cout << RoznicaKwadrat(x, y);
13     return 0;
14 }
```

Definicja funkcji

Wywołanie funkcji

Środowisko programowania dostępne bezpłatnie w Internecie:

Kompilator C++ <https://sourceforge.net/projects/orwelldevcpp/>

Podręczniki zawierające tematy z programowania w języku C++:

- G. Koba, *Z nowym bitem. Informatyka dla szkół ponadgimnazjalnych – zakres podstawowy*,
- G. Koba, *Informatyka dla szkół ponadgimnazjalnych – zakres podstawowy*,
- G. Koba, *Informatyka dla szkół ponadgimnazjalnych – zakres rozszerzony*.

Poradniki metodyczne:

- G. Koba, *Poradnik metodyczny. Informatyka dla szkół ponadgimnazjalnych – zakres podstawowy*,
- G. Koba, *Poradnik metodyczny. Informatyka dla szkół ponadgimnazjalnych – zakres rozszerzony*.

Materiały metodyczne w strefie nauczyciela: <https://nauczyciel.migra.pl/>

Przydatne linki: strona dotycząca programowania: <http://www.migra.pl/programowanie>
i strona autorska: <http://www.grazynakoba.pl>