

Введение в

C++

Урок 1-2

```
#include <iostream>
using namespace std;
void main()
{
int a,b;
cout << "Hello World!" << endl;
cout <<"Enter two numbers" << endl;
cin >>a >>b;
cout << "sum:" << a+b << endl;
system("pause");
}
```

**#include <iostream>** - директива процессора, предназначена для включения в исходный текст содержимое заголовочного файла, имя которого **<iostream>**, содержащий описания функций стандартной библиотеки ввода/вывода для работы с клавиатурой и экраном.

По простому: Без этой строчки не будут работать функции для вывода текста на экран И ввода с клавиатуры. Писать обязательно во всех программах.

**using namespace std;** - директива означает что, все определённые ниже имена будут относиться к пространству имён *std*.

По простому: обязательно писать и не думать зачем это надо.

**void main()** – основная функция программы, **void** означает что она не возвращает  
{  
    никаких значений  
Здесь находится собственно программа, между фигурных скобок.  
}

**Int a,b;** - объявление двух переменных типа **Int** – целочисленные переменные

**cout << " Enter two numbers" << endl;** - оператор вывода данных на экран ,  
    << - операция помещения данных в выходной поток;  
    **endl** - манипулятор, переводит сообщение на новую точку.

**cin >> a >> b;** - оператор ввода данных с клавиатуры, >> - операция для извлечения данных из выходного потока, читает значения из **cin** и сохр. их в переменных.

**cout <<"Sum:"<<a+b;** - оператор вывода, выводит на экран сумму 2х введенных чисел.

**system("pause");** - *системная команда пауза, ожидает нажатие любой клавиши, нужна для того что бы мы успели увидеть результат работы программы.*

**ВАЖНО!**

После любого оператора ставится точка с запятой.

Исключение: фигурные скобки.

После и перед фигурными скобками точка с запятой не ставится.

## Основные типы переменных

**Bool** – двоичная переменная принимает значения true(истина) или false(ложь).

**Char** – один символ.

**Short** – Целое число 2 байта. (от -32768 до 32767)

**Int** – Целое 4 байта(под 32х разрядной ОС) ( $\sim -2 \cdot 10^9$  до  $2 \cdot 10^9$ )

**Float** – Дробное 4 байта

**Double** – Дробное 8 байт

**Long Double** – Дробное 10 байт

Перед типом можно поставить идентификатор **unsigned** что означает без знака, тогда значения переменных будут только положительны и следовательно положительный диапазон увеличится в 2 раза.

*Арифметические операции: умножение (\*), деление (/), остаток от деления (%), сложение (+), вычитание (-).*

формат операция простого присваивания (=):

операнд\_1 = операнд\_2

пример: a=b=c=100, это выражение выполняется справа налево, результатом выполнения c=100, является число 100, которое затем присвоится переменной b, потом a. Все 3 переменные будут равны 100.

Сложные операции присваивания:

(\* =) – умножение с присвоением,

( / =) - деление с присвоением

(% = ) - остаток от деления с присвоением

(+ =) – сложение с присвоением

(- =) – вычитание с присвоением

пример: к операнду \_1 прибавляется операнд\_2 и результат записывается в операнд\_1

т.е.  $x = x + y$ , тогда **компактная запись**  $x += y$

*Операции увеличения (декремента) и уменьшения (инкремента)*

*на 1 (++ и --);*

пример:  $x = x + 1$  **компактная запись**  $x++$

# Оператор ветвления If

Синтаксис:

Сокращенная форма:

**If** (*условие*) выполняемое действие;

Если надо выполнить несколько действий тогда так:

**If** (*условие*)

{

*выполняемое действие*;

*выполняемое действие*;

}

Полная форма:

**If** (*условие*) выполняемое действие; **else** выполняемое действие;

**if (B) S1 ; else S2;**

При выполнении полной формы оператора **if** сначала вычисляется выражение **B**, затем анализируется его результат: если **B** истинно, то выполняется оператор **S1**, а оператор **S2** пропускается; если **B** ложно, то выполняется оператор **S2**, а **S1** - пропускается.

Так же вместо **else** можно написать **elseif** (*условие*) и задать еще одно условие, потом снова написать **elseif** и задать 3-ие условие и так сколько угодно раз.

*Операции отношения: (<, <=, >, >=, ==, !=), меньше, меньше или равно, больше, больше или равно, равно, не равно соответственно).  
Результатом операций являются значения true, false.*

## *Логические операции (&& и ||)*

***И (&&)** - возвращает значение истина тогда и только тогда, когда оба операнда принимают значение истина, в противном случае операция возвращает значение ложь.*

***ИЛИ (||)** - возвращает значение истина тогда и.т. тогда, когда хотя бы один операнд принимает значение истина, в противном случае – ложь*

*-логические операции выполняются слева направо;  
-приоритет операции && выше ||.*

**Пример:** Найдем большее из 2х чисел и выведем его на экран:

**Int x,y;**

**Cin >>x>>y;**

**If (x<y) cout << "Max=" << y;**

**elseif (x>y) cout << "Max=" << x;**

**else cout << "x=y";**

*Открою секрет любое число не равное 0 воспринимается как истина а 0 как ложь.  
Например If(5) – будет всегда выполняться.*