

**5 золотых правил
для тех, кто изучает
языки
программирования**

1. «Идеального»

языка

программирования

не существует

**2. Язык учат для
того, чтобы
применять
на практике**

**3. Все начинали
с нуля и когда-то
ничего не знали —
это нормально**

**4. Копировать,
экспериментироват
ь с кодом
и не переставать
учиться**

**5. Опубликовать
свою работу
и найти наставника**

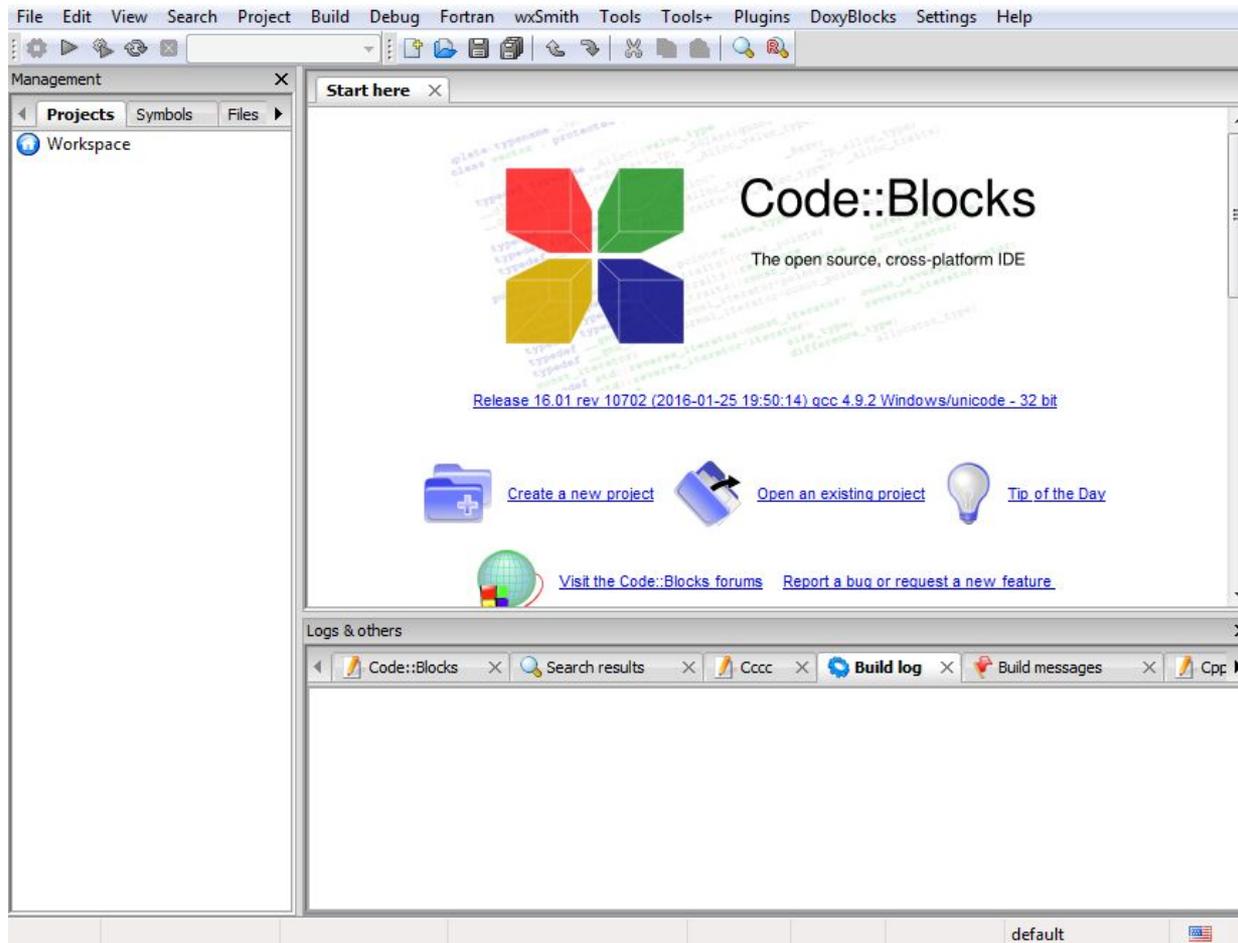
В качестве среды программирования на данном курсе выбрана **IDE Code::Blocks**, свободно распространяемая в сети Интернет. Все примеры будут приводится с использованием этой оболочки. Для успешного овладения навыками программирования, рекомендуется установить данную программу на свой компьютер.



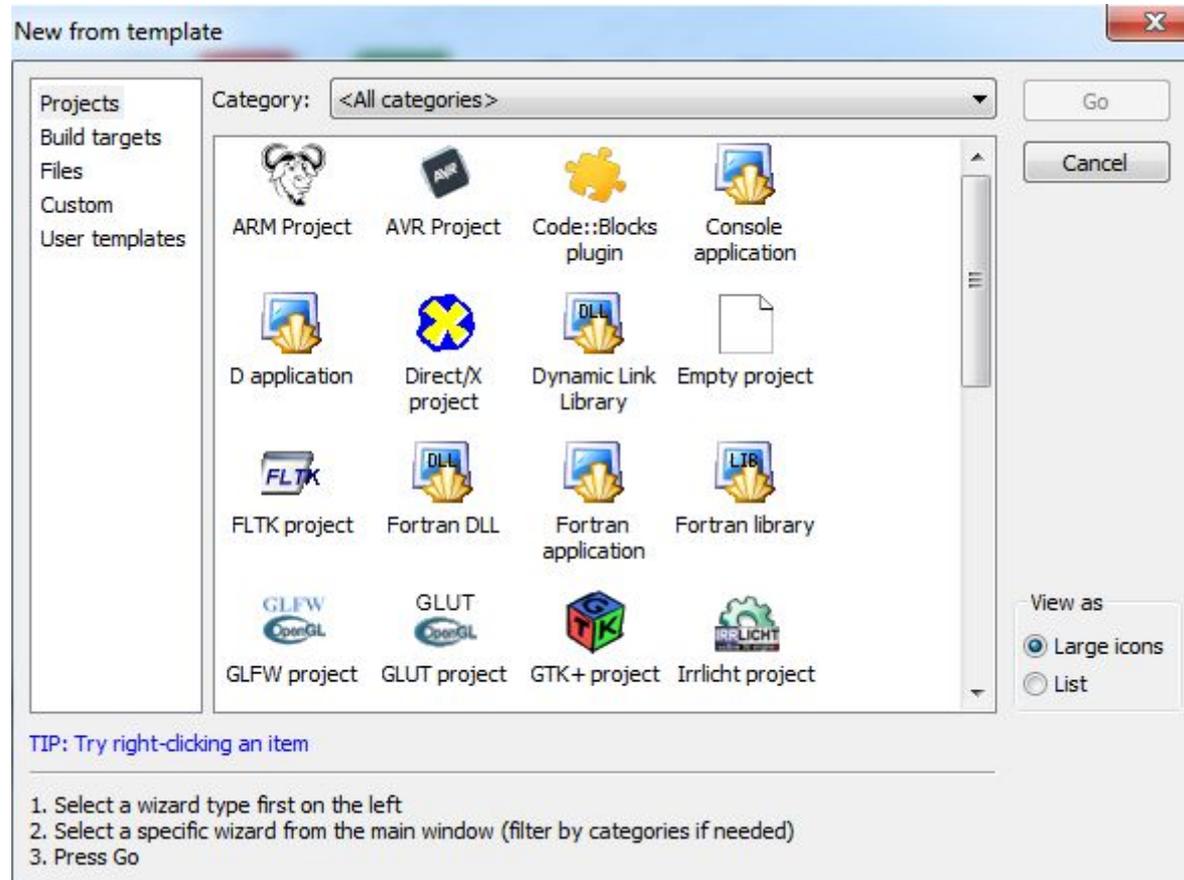
УСТАНОВКА КОМПИЛЯТОРА

- Для начала необходимо скачать дистрибутив данной программы. Для этого надо пройти на сайт <http://codeblocks.org/> и в меню слева **Main/Downloads/Binaries** выбрать **codeblocks-17.12mingw-setup.exe** с зеркала Sourceforge.net.

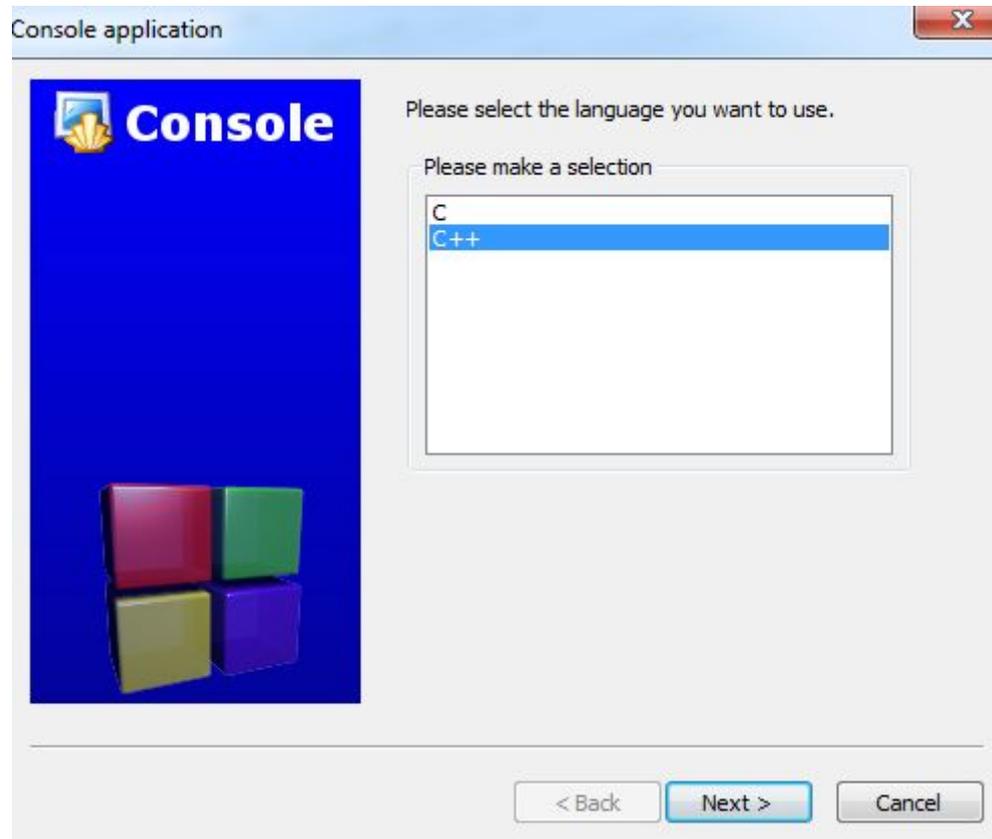
После того как дистрибутив скачан, запустите файл на выполнение. Нажимайте в диалоговых окнах **Next/Next/Next/Install**, оставляя все настройки по умолчанию. После запуска **Code::Blocks** должно появиться следующее окно:



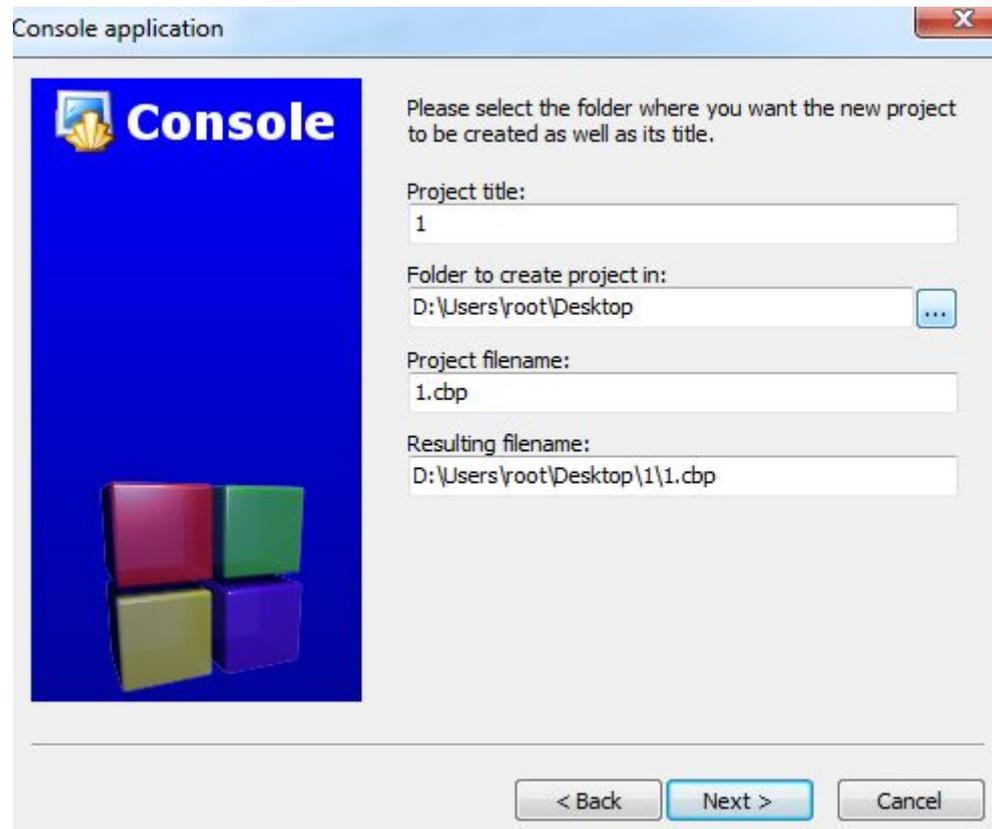
Выберите пункт **Create a New Project** (Создать Новый Проект), далее **Console Application**:



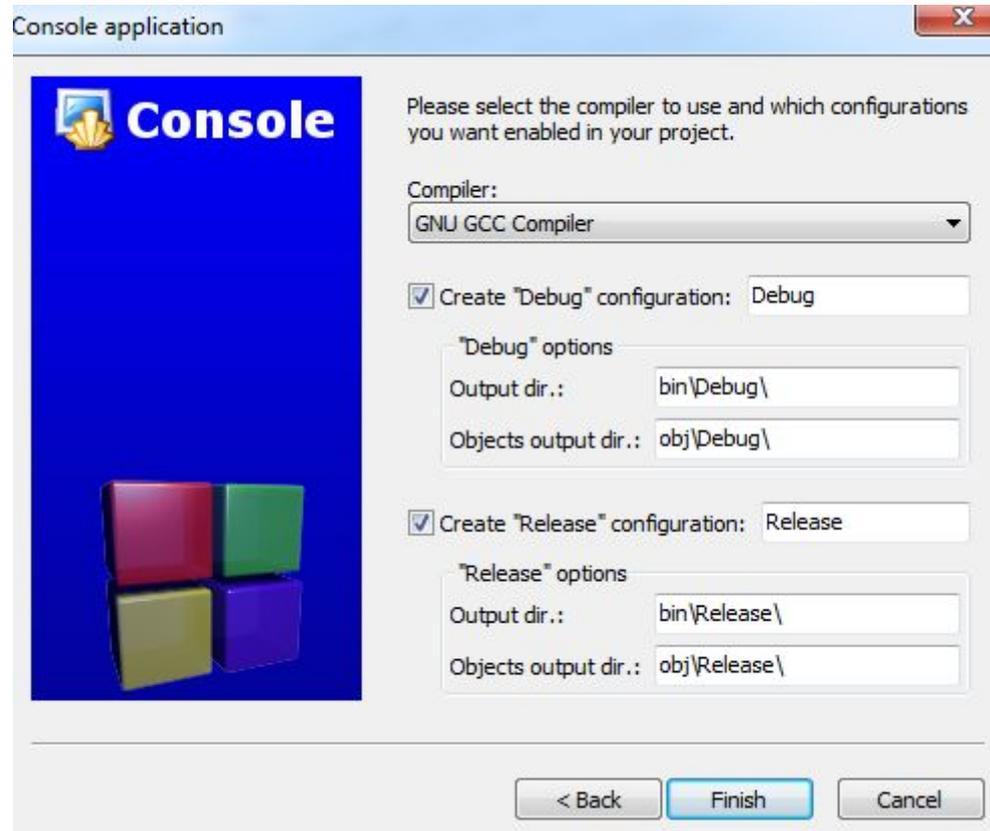
Далее выбираем C++ и Next:



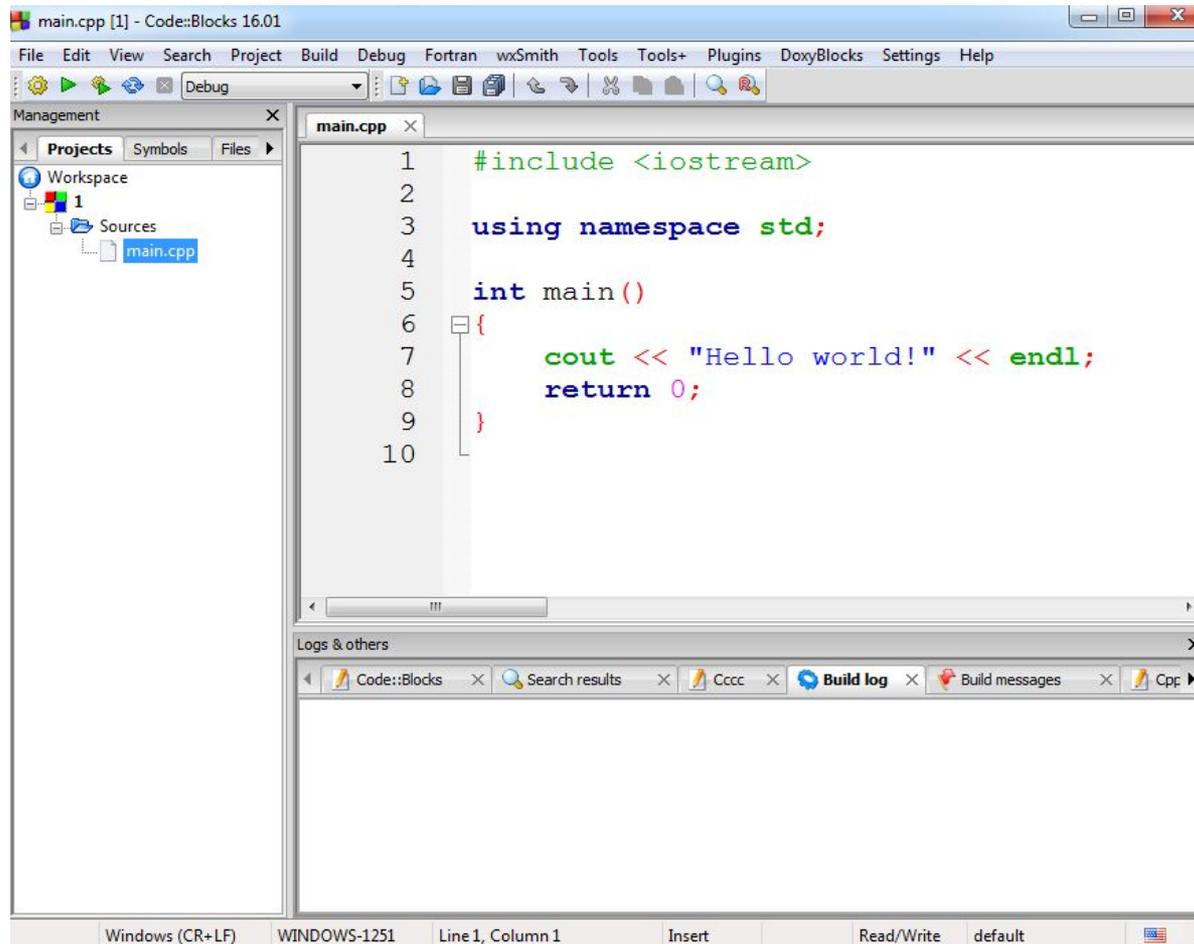
В следующем окне назовите свой проект (Project title), например 1, путь (Folder to create project in) -- Рабочий СТОЛ:



Следующее окно оставляем без изменения, нажимаем **Finish**.

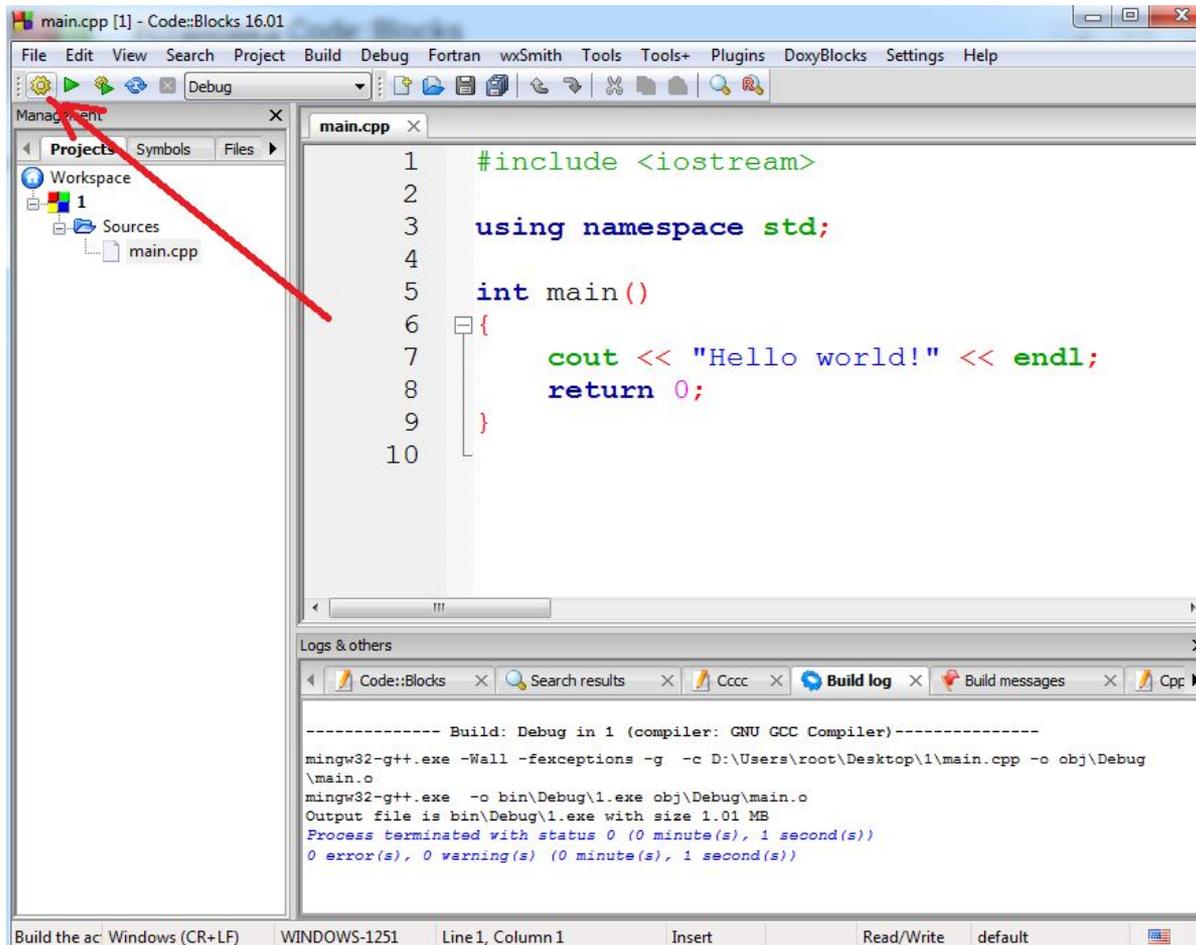


Раскройте ветвь **Sources** в левом меню, нажав на плюс.
Нажмите два раза на **main.cpp**. Справа, в окне редактора
откроется шаблон программы, который мы будем
модифицировать:

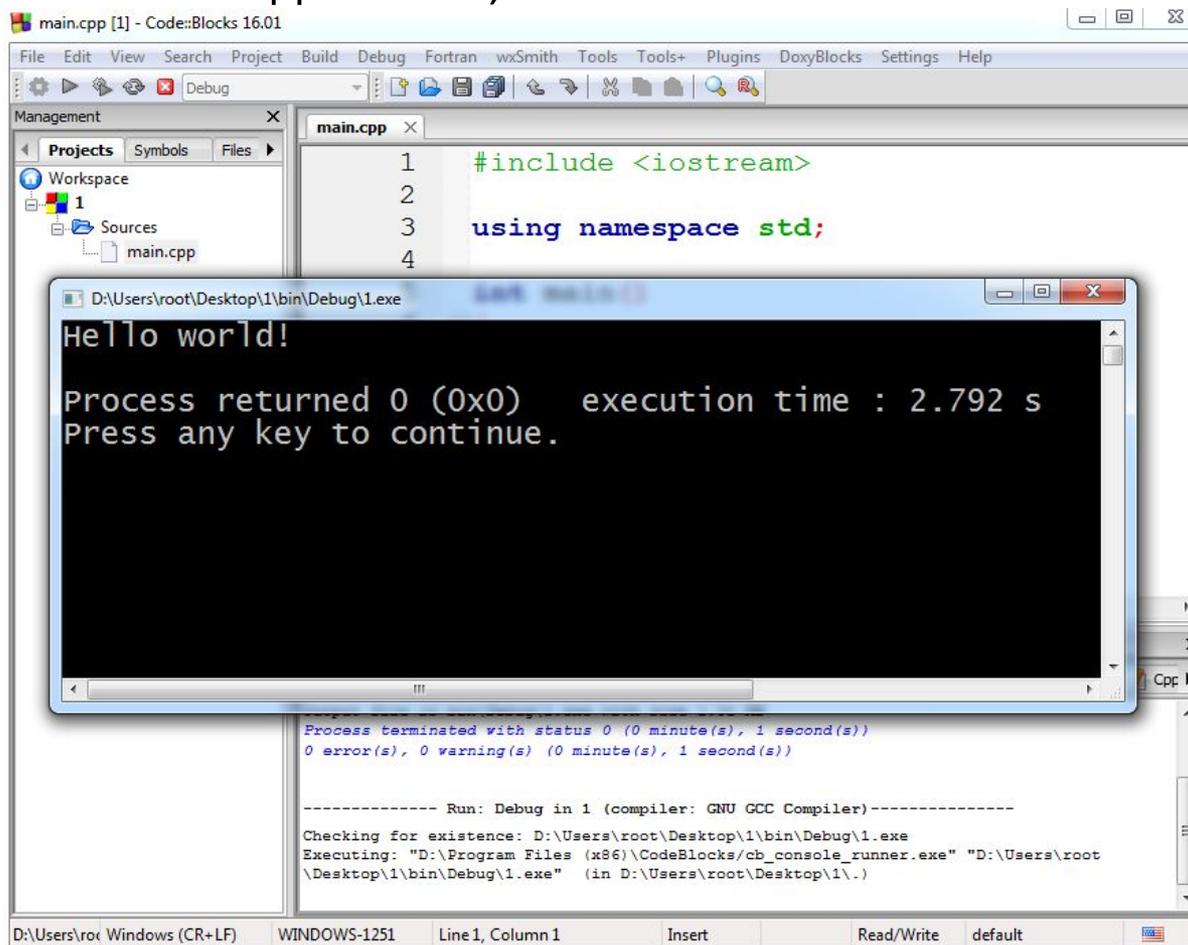


```
main.cpp [1] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Symbols Files
Workspace
1
Sources
main.cpp
main.cpp x
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     cout << "Hello world!" << endl;
8     return 0;
9 }
10
Logs & others
Code::Blocks x Search results x Cccc x Build log x Build messages x Cpp
Windows (CR+LF) WINDOWS-1251 Line1, Column1 Insert Read/Write default
```

Нажмите на кнопку **Build** (Собрать). Если в нижнем окне **Build log** выдаёт ошибки (не 0 errors), значит надо исправить ошибки и снова попытаться собрать проект.



Выполните программу нажатием на кнопку **Run** (рядом с **Build**). Должна открыться **Консоль**. Закрывать её можно нажатием на любую клавишу. Большинство наших программ будут запрашивать входные данные (числа, текст) с **Консоли** и выдавать результат в виде чисел, текста на **Консоль**.



ВНИМАНИЕ!!!!

**Чтобы наша программа
компилировалась, необходимо
назначить (выбрать) компилятор и
дебаггер по умолчанию**

Settings-Compiler->Reset Defaults

Settings-Debugger->Default

**и выбрать файл gdb32.exe из папки
C:\Program Files (x86)\CodeBlocks\MinGW\bin**

На этом уроке вам предстоит решить несколько задач на языке C++. Для создания программы на C++ вам понадобится следующий шаблон, который необходимо переписать в тетрадь и выучить наизусть. Данная программа выполнится, но ничего не попросит ввести и ничего не выведет.

```
#include <iostream>
using namespace std;
int main() {
    // your program
    return 0;
}
```

Укажите правильное написание строки заголовка в программе C++

- `#include <iostream>`
- `#include <iostream>;`
- `#include <iostrem>`
- `#includ <iostream>`

Укажите правильное написание строки пространства имён

- `using namespace std;`
- `use namespace std;`
- `using namespacing std;`
- `uses namespace std;`

Добавим строки ввода и вывода данных. Данная программа при запуске откроет консоль и будет ожидать, пока пользователь не наберёт целое число. После нажатия клавиши **Enter** число будет размещено в переменной `aa`. Далее оно увеличится на единицу и будет выведено обратно на консоль.

```
#include <iostream>
using namespace std;
int main() {
    int a;
    cin >> a;
    a++;
    cout << a;
    return 0;
}
```

- В программах можно использовать следующие арифметические операции:
- Сложение + (знак плюс);
- Вычитание - (знак минус);
- Умножение * (знак звёздочка);
- Деление / (знак косая черта, слэш);
- Остаток от деления % (знак процент)
- При возведении в степень следует использовать умножение, например:
- $a^2 = a * a$
- $a^3 = a * a * a$
- $(a+b)^4 = (a+b) * (a+b) * (a+b) * (a+b)$

Напишите программу, которая запрашивает натуральное число и выводит его квадрат.

```
#include <iostream>
```

```
int main() {
```

```
// тут вы должны написать код
```

```
    return 0;
```

```
}
```

Данная программа при запуске откроет консоль и будет ожидать, пока пользователь не наберёт два целых числа, разделённых пробелом или переводом строки. После нажатия клавиши **Enter** первое число будет размещено в переменной **a**, а второе число -- в переменной **b**. Далее будет выведена на консоль сумма этих двух чисел.

```
#include <iostream>
using namespace std;
int main() {
    int a, b;
    cin >> a >> b;
    cout << a + b;
    return 0;
}
```

Вводятся две целочисленные переменные a и b .
Напишите программу, которая находит значение
выражения:

$$(a + b)^2 + (a - b)^2 - 3ab$$

Ввод 1 2

Вывод 4

Вводятся две целочисленные переменные a и b .
Напишите программу, которая находит значение
выражения:

$$(a^2 - b)^2 - 4ab^2 + 4(a^3 - 5)^2.$$

Ввод 3 2
Вывод 1937

Данная программа запрашивает три целых числа a,b,c. Программа в первой строке выводит сумму a+b, во второй строке a+c, в третьей строке b+c, в четвёртой, через пробел сумму всех трёх переменных и их произведение.

```
#include <iostream>
using namespace std;
int main() {
    int a, b, c;
    cin >> a >> b >> c;
    cout << a + b << endl;
    cout << a + c << endl;
    cout << b + c << endl;
    cout << a + b + c << " " << a * b * c << endl;
    return 0;
}
```

Напишите программу, которая запрашивает два целых числа, и выводит их в обратном порядке через пробел.

Ввод 3 2
Вывод 2 3

Напишите программу, которая запрашивает четыре
целых числа и выводит их в обратном порядке по
одному в строке.

Ввод 1 2 3 4

Вывод

4

3

2

1

Напишите программу, которая запрашивает три цифры (от 0 до 9) и выводит число, получающееся из этих цифр в том же порядке, что и при вводе.

Ввод 1 2 3

Вывод 123

УСЛОВНЫЙ ОПЕРАТОР

Данная программа вводит натуральное число, и если оно чётное пишет YES, если нечётное -- NO.

```
#include <iostream>
using namespace std;
int main() {
    int a;
    cin >> a;
    if (a % 2 == 0) { cout << "YES"; } else { cout << "NO"; }
    return 0;
}
```

Для сравнения чисел можно использовать следующие операции:

Равно == (два знака равно подряд);

Не равно != (восклицательный знак и знак равно подряд);

Больше >;

Меньше <;

Больше или равно >= (знак больше и знак равно подряд);

Меньше или равно <= (знак меньше и знак равно подряд).

Напишите программу, которая запрашивает целое число, и если оно положительное -- пишет "YES", в противном случае -- "NO".

```
#include <iostream>
```

```
int main() {
```

```
// тут вы должны написать код
```

```
    return 0;
```

```
}
```

**Напишите программу, которая запрашивает целое число, и если оно меньше или равно 1000 -- пишет "YES",
в противном случае -- "NO".**

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
// тут вы должны написать код
```

```
return 0;
```

```
}
```

ВЕТВЛЕНИЕ

Данная программа запрашивает натуральное число n и выводит на экран n пятёрок.

```
#include <iostream>
using namespace std;
int main() {
    int n, i;
    cin >> n;
    for (i = 0; i < n; i++) { cout << 5; }
    return 0;
}
```

Напишите программу, которая запрашивает натуральное число N (не более 50), и выводит два ряда из N звёздочек.

```
#include <iostream>
using namespace std;
int main() {
    // тут вы должны написать код
    return 0;
}
```