

C++ Notes

Part 1-1

**Slides were created from
CodewithMosh.com**

[All](#) [Videos](#) [Images](#) [Short videos](#) [Shopping](#) [Forums](#) [News](#) [More](#)

GDB online Debugger
https://www.onlinegdb.com/online_c++_compiler :

Online C++ Compiler - online editor

Online C++ Compiler. Code, Compile, Run and Debug C++ program online. Write your code in this editor and press "Run" button to compile and execute it.



The screenshot shows the OnlineGDB IDE interface. The top navigation bar includes links for Getting Started, Most Visited, Isidore : University ..., HT SIGORTA K..., Mike Zellers - ..., Jeongkyu Lee - You..., G on'live hotel çeşme ..., Anasayfa - Devletin ..., and a search bar with the query "K". The main menu on the left lists IDE, My Projects, Classroom (new), Learn Programming, Programming Questions, Sign Up, and Login. The central workspace shows a code editor with the file "main.cpp" open. The code is:

```
1 //*****
2 //***** Online C++ Compiler.
3 //***** Code, Compile, Run and Debug C++ program online.
4 //***** Write your code in this editor and press "Run" button to compile and execute it.
5 //*****
6
7 #include <iostream>
8
9 int main()
10 {
11     std::cout<<"Hello World";
12
13     return 0;
14
15 }
```

C++ Online IDE

Code::BLOCKS

The IDE with all the features you need, having a consistent look, feel and operation across platforms.

```
24     if (con_callback)
25     {
26         con_callback();
27     }
28     if (discon_callback)
29     {
30         discon_callback();
31     }
32     if (!rx_callback((const char*)buf, len))
33     {
34         if (!rx_callback((const char*)buf, len))
35         {
36             return;
37         }
38         bool hasln = false;
39         uint32_t cnt = 0;
40         while (cnt < len)
41         {
42             char ch = buf[cnt++];
43             hasln |= ch == '\n';
44             if (rx.write(ch) == CircularBufferBase::kError)
45                 break;
46         }
47         bool full = rx.status() == CircularBufferBase::kFull;
48         if (rx_ln_callback && (hasln || full))
49         {
50             uint16_t lenOut;
51             rx.read_until((const uint8_t*)"\\n", (uint16_t*)&lenOut);
52             rx_ln_callback(lineBuffer, lenOut, full);
53         }
54     }
55 }
56 void USBDevice::setConnectCallback(CONNECT_CB callback)
57 {
58     con_callback = callback;
59 }
```

News
Features
Downloads
User manual
Forums
Wiki
License
Donations

GPL W3C css GetFirefox

SOURCEFORGE Support this project

Code::Blocks

Code::Blocks

The free C/C++ and Fortran IDE.

Code::Blocks is a free C/C++ and Fortran IDE built to meet the most demanding needs of its users. It is designed to be very extensible and fully configurable.

Built around a plugin framework, Code::Blocks can be extended with plugins. Any kind of functionality can be added by installing/coding a plugin. For instance, event compiling and debugging functionality is provided by plugins!

If you're new here, you can read the [user manual](#) or visit the [Wiki](#) for documentation. And don't forget to visit and join our [forums](#) to find help or general discussion about Code::Blocks.

We hope you enjoy using Code::Blocks!

The Code::Blocks Team

Latest news

Code::Blocks 25.03 is here!

Finally, after quite some time, we are back with many improvements, new features, more stable, further developments for HIDAPI, and Code::Blocks has never been better!

[All](#) [Videos](#) [Images](#) [Short videos](#) [Forums](#) [Shopping](#) [Web](#) [More](#)

SourceForge

<https://sourceforge.net> › ... › Dev-C++

⋮

Dev-C++ - Browse /Compilers/MinGW at SourceForge.net

... Download Latest Version Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup.exe (50.4 MB). Email in envelope. Get

an email when there's a new version of Dev-C++. Next. Home ...

The screenshot shows the SourceForge website for the Dev-C++ project. At the top, there's a navigation bar with links for Business Software, Open Source Software, SourceForge Podcast, Resources, and a search bar. A banner for SOC Radar is visible. On the left, there's a sidebar with links for Getting Started, Most Visited, and other projects like Isidore : University ... and SIGORTA K... Below the sidebar, the main content area features a large image of the Dev-C++ IDE icon and the text "Dev-C++ Files: A free, portable, fast and simple C/C++ IDE Brought to you by: orwelldevcpp". Below this, there are tabs for Summary, Files (which is selected), Reviews, Support, External Link ▾, Tracker, Code, and Forums. A prominent green button allows users to "Download Latest Version Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup.exe (50.4 MB)". To the right of this button is a form to "Get an email when there's a new version of Dev-C++" with fields for "Enter your email" and "Next". A feed icon is also present. The main table lists four versions of MinGW: MinGW 4.8.1.7z (2014-02-13, 36.7 MB, 24 downloads/week), MinGW 4.7.2.7z (2013-01-13, 14.8 MB, 2 downloads/week), MinGW 4.7.0.7z (2012-09-28, 14.9 MB, 1 download/week), and MinGW 4.6.2.7z (2012-09-28, 14.1 MB, 1 download/week). At the bottom, it says "Totals: 4 Items" and "80.6 MB" with a total download count of 28. To the right of the main content, there's a sidebar titled "Recommended Projects" listing Code::Blocks, Arduino, Apache OpenOffice, KeePass, and DeSmuME: Nintendo DS emulator.

Name	Modified	Size	Downloads / Week	Actions
MinGW 4.8.1.7z	2014-02-13	36.7 MB	24	Download
MinGW 4.7.2.7z	2013-01-13	14.8 MB	2	Download
MinGW 4.7.0.7z	2012-09-28	14.9 MB	1	Download
MinGW 4.6.2.7z	2012-09-28	14.1 MB	1	Download

C++ IDEs

Popular IDEs

- MS Visual Studio

<https://visualstudio.microsoft.com/vs/community/>

- XCode

App Store

- CLion

<https://www.jetbrains.com/clion/download>



JetBrains

<https://www.jetbrains.com/clion/>

CLion: A Cross-Platform IDE for C and C++ by JetBrains

A single IDE for all your C and C++ needs. No matter how you use C and C++, CLion helps make your development experience smoother and more productive.

The screenshot shows the official website for CLion, a cross-platform IDE for C and C++. The page features a dark background with abstract, overlapping colored shapes in shades of green, blue, and purple. In the top left corner, the JetBrains logo is displayed next to the word "CLion". Below the logo, the text "Free for non-commercial use" is shown in a small green button. The main headline reads "A cross-platform IDE for C and C++". Below the headline, the tagline "Harness the power. Cut the complexity." is visible. At the bottom of the page, there is a prominent "Download" button. The top navigation bar includes links for AI, Developer Tools, Team Tools, Education, Solutions, Support, Store, and a search bar. The right side of the navigation bar has buttons for Pricing and Download. The address bar at the top of the browser window shows the URL <https://www.jetbrains.com/clion/>.

C++ Overview

□ High Performance & Efficiency

C++ remains one of the oldest yet most popular programming languages thanks to its speed and resource efficiency.

□ Widely Used in Critical Applications

Ideal for performance-intensive software such as:

- Video games (e.g., Unreal Engine)
- Server applications
- Operating systems

□ Learning Path

To master C++, focus on:

- Syntax and grammar
- The C++ Standard Library (pre-written solutions for common problems)

C++ Overview

□ Development Tools

Most C++ applications are built using IDEs such as:

- Microsoft Visual Studio
- XCode
- CLion

□ Compilation Process

Before running a C++ program, the source code must be compiled into machine code.

□ Program Entry Point

Every C++ application starts with the main() function.

Main Function

```
#include <iostream>
int main() {
    std::cout << "Hello World";
    return 0;
}
```



Input/Output Standard Library

Naming Conventions

```
int fileSize; // Snake Case
● int FileSize; // Pascal Case
int fileSize; // Camel Case
int iFileSize; // Hungarian Notation
return 0;
```

Variables and Constants

```
#include <iostream>

int main() {
    const double pi = 3.14;
    pi = 0;
    return 0;
}
```

Arithmetic Operations

```
int main() {
    int x = 10;
    // int y = x++; // x = 11, y = 10
    int z = ++x; // x = 11, z = 11
    std::cout << z;
    return 0;
}
```

Arithmetic Operations

```
#include <iostream>

int main() {
    // ()
    // * and /
    // + and -
    double x = (1 + 2) * 3;
    std::cout << x;
    return 0;
}
```

Output

```
#include <iostream>

int main() {
    int x = 10;
    std::cout
    return 0;
}
```

Standard Output Stream

Console or terminal

Set of characters

Output

```
#include <iostream>

int main() {
    int x = 10;
    int y = 20;
    std::cout << "x = " << x << std::endl
                  << "y = " << y;
    return 0;
}
```

Stream Insertion Operator

Output

```
#include <iostream>

using namespace std;

int main() {
    int x = 10;
    int y = 20;
    cout << "x = " << x << endl
        << "y = " << y;
    return 0;
}
```

Namespace

In C++, `std` is an abbreviation for "standard" and refers to the standard namespace. This namespace encompasses the C++ Standard Library, which provides a wide range of functionalities, including:

- **Data Structures:** `std::vector`, `std::list`, `std::array`, `std::map`, etc.
- **Input/Output:** `std::cout`, `std::cin`.
- **Algorithms:** `std::sort`, `std::find`, `std::copy`.
- **Function Objects:** `std::function`.
- **Strings:** `std::string`.
- **Utilities:** such as exception handling, memory management, and more.

Namespace

The `std` namespace serves to organize these library components and prevent naming conflicts with user-defined code or other libraries. To access elements within the `std` namespace, one can either use the scope resolution operator (`::`) or the `using namespace std;` directive. For example:

```
#include <iostream>

int main() {
    std::cout << "Hello, world!" << std::endl; // Using scope resolution op

    return 0;
}
```

OR

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

int main() {
    cout << "Hello, world!" << endl; // Using namespace directive
    return 0;
}
```

Variable Names

- The variable name should begin with an alphabet.
- Digits may be used in the variable name but only after the alphabet.
- No special symbols can be used in variable names except for the underscore('_').
- No keywords can be used for variable names.

Input

```
cout << "Enter values for x and y: ";
double x;
double y;
cin| >> x >> y;
cout << x + y;
return 0;
}
```

Stream Extraction Operator

Example

```
int main() {
    cout << "Fahrenheit: ";
    int fahrenheit;
    cin >> fahrenheit;
    double celsius = (fahrenheit - 32) / 1.8;
    cout << celsius;
    return 0;
}
```

Example

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

using namespace std;

int main() {
    double result = floor(1.2);           I
    cout << result;
    return 0;
}
```

Example

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

using namespace std;

int main() {
    double result = pow( lcpp_x: 2, lcpp_y: 3);
    cout << result;
    return 0;
}
```

Example

```
int main() {  
    cout << "Enter radius: ";  
    double radius;  
    cin >> radius;  
    const double pi = 3.14;  
    double area = pi * pow( lcpp_x: radius, lcpp_y: 2 );  
    cout << area;  
    return 0;  
}
```

Comments

```
int main() {  
    // Declare a variable and initialize to 0  
    int x = 0;  
    return 0;  
}
```

Single line comment

```
int main() {  
    /*  
     *  
     */  
    int x = 0;  
    return 0;
```

Multiple line comment

Fundamental Data Types

STATICALLY-TYPED

- C++
- C#
- Java

DYNAMICALLY-TYPED

- Python
- JavaScript
- Ruby

Type Casting

- **Type Casting:** It allows you to treat a variable of one type as another type for a specific operation. However, it does not change the original variable's type.

C++

```
int intValue = 10;  
double doubleValue = (double)intValue; // Type casting int to double
```

- **Creating a new variable:** You can create a new variable of the desired type and assign the value of the original variable to it.

C++

```
int intValue = 10;  
double doubleValue = intValue; // Implicit conversion to double
```

Fundamental Data Types

Whole Numbers

Type	Bytes	Range
short	2	-32,768 to 32,767
int	4	-2B to 2B
long	4	Same
long long	8	

Fundamental Data Types

Numbers with Decimal Places

Type	Bytes	Range
float	4	-3.4E38 to 3.4E38
double	8	-1.7E308 to 1.7E308
long double	8	-3.4E932 to 1.7E4832

Fundamental Data Types

Type	Bytes	Range
bool	1	true / false
char	1	

Example

```
#include <iostream>

using namespace std;

int main() {
    double price = 99.99;
    float interestRate = 3.67f;
    long fileSize = 90000L;
    char letter = 'a';
    bool isValid = false;
    return 0;
}
```

```
int main() {
    auto price : double = 99.99;
    auto interestRate : double = 3.67;
    auto fileSize = 90000L;
    auto letter = 'a';
    auto isValid = false;
    return 0;
}
```

Brace Initialization

```
int main() {  
    int number {};  
    cout << number;  
    return 0;  
}
```

Number is initialized to zero. Otherwise, an unknown value.

Unsigned int – be careful!

```
int main() {  
    unsigned int number = 0;  
    number--;  
    cout << number;  
    return 0;  
}
```

Narrowing Numbers

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

int main() {
    int number = 1'000'000;
    short another = number;
    return 0;
}
```

The value is never used
Clang-Tidy: Narrowing conversion from 'int' to signed type 'short' is implemented
Remove initializer ⌂ More actions... ⌂

```
int number = 1'000'000
```

Narrowing Numbers

```
int number = 1'000'000;
short another{number};
cout << another;
return 0;
}
```

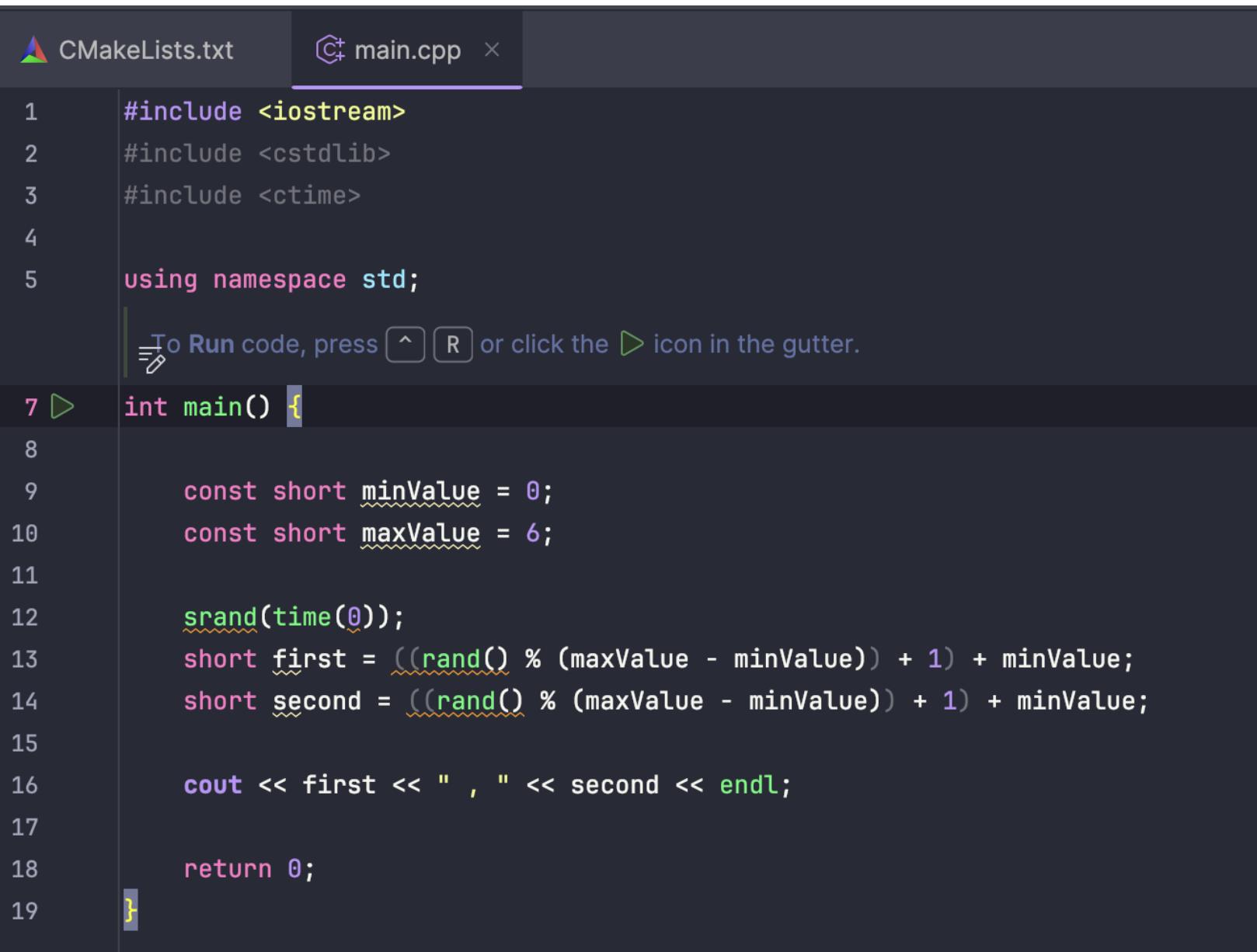
Use brace initializer {}

Random Number Generation

```
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main() {
    srand(time(0));
    int number = rand() % 10;
    cout << number;
    return 0;
}
```



A screenshot of a code editor showing a C++ file named `main.cpp`. The code implements a simple program to roll two dice. It includes headers for `<iostream>`, `<cstdlib>`, and `<ctime>`. It uses the `rand()` function from `cstdlib` to generate random numbers between 1 and 6. The `time(0)` function from `ctime` is used to seed the random number generator. The program outputs the results of the first and second rolls to the console.

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4
5 using namespace std;
6
7 int main() {
8
9     const short minValue = 0;
10    const short maxValue = 6;
11
12    srand(time(0));
13    short first = ((rand() % (maxValue - minValue)) + 1) + minValue;
14    short second = ((rand() % (maxValue - minValue)) + 1) + minValue;
15
16    cout << first << " , " << second << endl;
17
18    return 0;
19 }
```

Example: Rolling Dice