Beginning C++ Programming - From Beginner to Beyond

David Corzo

2020 May 18

Contents

Chapter 1

Introduction

1.1 Why learn C++?

- Popular:
  - Lots of code is still written in C++.
  - Programming language popularity indexes ranks C++ high.
  - Active community, GitHub, Stack overflow.
- Relevant:
  - Windows, Linux, MaxOSX, Photoshop, Illustrator, MySQL, MongoDB.
  - Amazon, Apple, Microsoft, PayPal, Google, Facebook, MySQL, Oracle, HP, IBM, more...
  - VR, Unreal Engine, Machine learning, networking & telecom, more...
- Powerful:
  - Super-fast, scalable, portable.
  - Supports both procedural and object-oriented programming.
- Good career opportunities:
  - C++ skills always in demand.
  - C++ = Salary++.

1.2 Modern C++ and the C++ standard

- Early 1970s: C programming language; Dennis Ritchie.
- 1979: Bjarne Stroustrup; 'C with classes'.
- 1983: Name changed to C++.
- 1989:First commercial release.
- 1998: C++98 Standard.
- 2003: C++03 Standard.
- 2011: C++11 Standard.
- 2014: C++14 Standard.
- 2017: C++17 Standard.

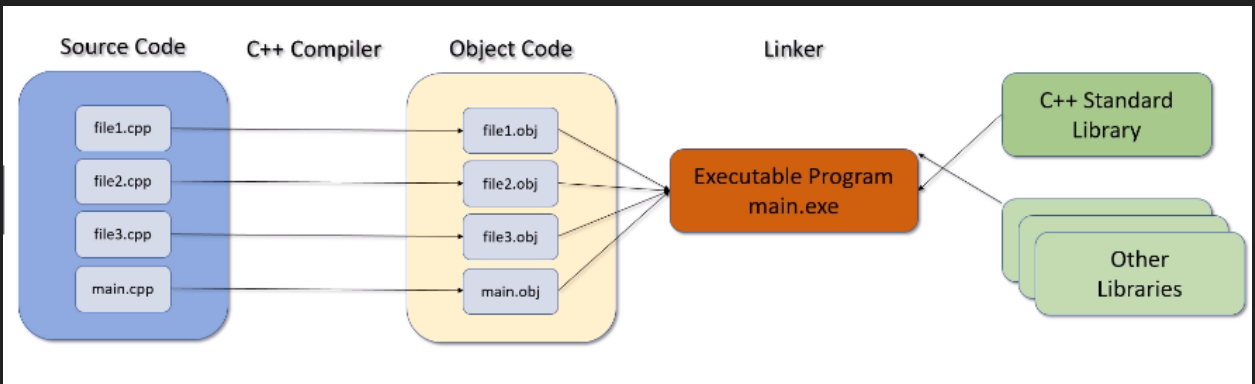
1.2.1 Modern C++ and C++ Standard

- Classical C++: Pre C++11 Standard.
- Modern C++:
  - C++11: Lots of new features.
  - C++14: Smaller changes.
  - C++17: Simplification.

1.3 How does it all work?

- Use non-ambiguous instructions.
- Programming language: source code, high level, for humans.
- Editor: text editor. *.cpp* and *.h* files.
- Binary or other low level representation: object code for computers.
- Compiler: Translates from high-level to low-level.
- Linker: links together our code with other libraries, creates *.exe*.
- Testing and debugging: finding and fixing program errors.

1.3.1 The C++ build process



1.3.2 Integrated Development Environments (IDEs)

- Editor.
- Compiler.
- Linker.
- Debugger.
- Keep everything in sync.

IDEs

- CodeLite.
- Code::Blocks.
- NetBeans.
- Eclipse.
- CLion.
- Dev-C++.
- KDevelop.
- Visual Studio.
- Xcode.

Chapter 2

Installation and setup

2.1 Installing C++ Compiler for windows

- Go to: `http://mingw-w64.org/doku.php/download/mingw-builds`
- Go to: Downloads, find the build, download and run executable.
- Set the environment variable:
  - Control panel → Edit system environment variables.
  - Environment variables → System → Path → Edit.
  - New → Browse → < go to your instaltion dir > → OK.
- Go to CMD: type `c++ --version` → Should print version.

2.2 VSCode Project Setting Up

Inside the .vscode directory add:

- `c_cpp_properties.json`:

```
{
    "configurations": [
        {
            "name": "Win32",
            "includePath": [
                "${workspaceFolder}/**"
            ],
            "defines": [
                "_DEBUG",
                "UNICODE",
                "_UNICODE"
            ],
            "browse": {
                "path": [
                    "${workspaceRoot}",
                    "C:\\Program Files\\mingw-w64\\mingw64\\bin\\gcc.exe"
                ]
            },
            "compilerPath": "C:\\Program Files\\mingw-w64\\mingw64\\bin\\gcc.exe",
            "cStandard": "gnu18",
```

```
            "cppStandard": "gnu++14"
        }
    ],
    "version": 4
}
```

- launch.json:

```
{
    "version": "0.2.0",
    "configurations": [
        {
            "name": "g++.exe - Compilar y depurar el archivo activo",
            "type": "cppdbg",
            "request": "launch",
            "program": "${fileDirname}\\${fileBasenameNoExtension}.exe",
            "args": [],
            "stopAtEntry": false,
            "cwd": "${workspaceFolder}",
            "environment": [],
            "externalConsole": false,
            "MIMode": "gdb",
            "miDebuggerPath": "C:\\Program Files\\mingw-w64\\mingw64\\bin\\gdb.exe",
            "setupCommands": [
                {
                    "description": "Habilitar la impresión con sangría para gdb",
                    "text": "-enable-pretty-printing",
                    "ignoreFailures": true
                }
            ],
            "preLaunchTask": "C/C++: g++.exe build active file"
        }
    ]
}
```

- In order to establish the formatting style put the following in `settings.json`:

```
{
    "C_Cpp.clang_format_fallbackStyle": "{ BasedOnStyle: LLVM, UseTab: Never, IndentWidth: 4, TabWi
    "emmet.showSuggestionsAsSnippets": true,
    "files.associations": {
        "*.rmd": "markdown",
        "iostream": "cpp"
    }
}
```

- task.json:

```
{
    "version": "2.0.0",
    "tasks": [
        {
            "label": "C/C++: g++.exe build active file",
            "type": "shell",
            "command": "C:\\Program Files\\mingw-w64\\mingw64\\bin\\g++.exe",
            "args": [
```

```
                "-g", "main.cpp", "-o", "a.exe" // , "&&", "main"
            ],
            "problemMatcher": [
                "$gcc"
            ],
            "presentation": {
                "echo": false,
                "reveal": "always",
                "focus": false,
                "panel": "shared",
                "showReuseMessage": true,
                "clear": false
            },
            "group": {
                "kind": "build",
                "isDefault": true
            }
        }
    ]
}
```

Chapter 3

Curriculum Overview

3.1 Curriculum overview

Get started quickly programming in C++.

- Getting started.
- Structure of a C++ program.
- Variables and constants.
- Arrays and vectors.
- Strings in C++.
- Expressions, Statements and Operators.
- Statements and operators.
- Determining control flow.
- Functions.
- Pointers and references.
- OPP: Classes and objects.
- Operator overloading.
- Inheritance.
- Polymorphism.
- Smart pointers.
- The Standard Template Library (STL).
- I/O Stream.
- Exception Handling.

Chapter 4

Getting Started

4.1 Writting our first program

- Create a project.
- Create a file and type the following code.
- This program is going to take in a number and then display "Wow that is my favorite number".

```
#include <iostream>
int main() {
    int favorite_number; // stores what the user will enter.
    std::cout << "Enter your favorite number between 1 and 100"; // prints.
    std::cin >> favorite_number;
    std::cout << "Amazing!! That's my favorite number too!" << std::endl; // prints that line. endl adds
}
```

4.2 Building our first program

- Building involves compiling and linking.
- In vscode you can run the compile task by pressing ctrl+b.
- Linking means grabbing all the dependencies the main function needs, making .o or object files and adding them to the executable or the .exe.
- Tipically modern compilers have the option to not produce the object files and go ahead and just produce a single executable. IDEs tipically also hide the object files if they are produced.
- By *cleaning* a project what we mean is the object files are deleted and an executable will be produced.

4.3 What are compiler errors?

- Programming languages have rules.
- Syntax errors: something wrong with the structure.

  ```
  std::cout << "Errors << std::endl; // the string is never terminated.
  ```

- Semantic errors: something wrong with the meaning:

  ```
  a + b; // to sum a and b when it doesn't make sense to add them, maybe they are not numbers for exa
  ```

4.3.1 Examples of errors

Not enclosing a string with the ” characters.

```
int main() {
    std::cout << "Hello world << std::endl; // string is not terminated with the other ".
    return 0;
}
```

Typos in your program:

```
int main() {
    std::cout << "Hello world" << std::endll; // endll doesn't exist, this is syntaxis errors.
    return 0;
}
```

Missing semi-colons:

```
int main() {
    std::cout << "Hello world" << std::endl // missing semicolon.
    return 0;
}
```

Function doesn’t return the type specified, in this case the function doesn’t return an integer.

```
int main() {
    std::cout << "Hello" << std::endl;
    return; // main needs to return an integer and it is returning a void.
}
```

Not returning the specified type.

```
int main() {
    std::cout << "Hello World" << std::endl;
    return "Hello"; // "Hello" is not an integer. Error.
}
```

Missing Curly brace:

```
int main() // opening curly brace missing.
    std::cout << "Hello World" << std::endl;
}
```

Semantic error (example: adding something when it doesn’t make sense).

```
int main() {
    std::cout << ("Hello world" / 125) << std::endl; // dividing a string by a number, this doesn't mak
    return 0;
}
```

4.4 What are compiler warnings?

- It is good practice to never ignore compiler warnings.
- The compiler will recognize a potential issue but is still able to produce object code from the source code, things such as uninitialized variables.
- It’s only a warning because the compiler is still able to generate correct machine code.
- Example:

```cpp
int miles_driven; // never initialized, this value could be anything.
std::cout << miles_driven << std::endl;
/* Warning: 'miles_driven' is used uninitialized in this function. */
```

- Another example is when you declare variables and never use them.

```cpp
int miles_driven = 100;
std::cout << "Hello world" << std::endl;
/* Warning: unused variable 'miles_driven'. */
```

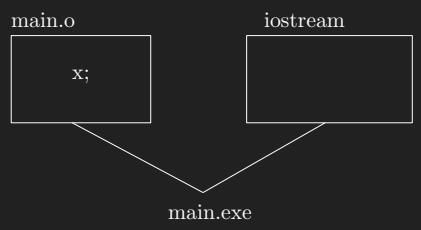
- As a rule you want to produce warning free source code.

4.5 Linked errors

- The linker is having trouble linking all the object files together to create an executable.
- Usually there is a library or object file that is missing.

4.5.1 Example

```cpp
#include <iostream>
extern int x; // this means the variable is defined outside this file.
int main() {
    std::cout << "Hello world" << std::endl;
    std::cout << x;
    return 0;
}
/* This program will compile, but in runtime you will get a linker error. */
```



According to the linker x is undefined, thus an error is produced.

4.6 Runtime Errors

- Errors that occur when the program is executing.
- Some typical runtime errors include:
  - Divide by zero.
  - File not found.
  - Out of memory.
- Can cause your program to crash.
- Exception handling can help deal with runtime errors.

4.7 What are logic errors?

- Errors or bugs in your code that cause your program to run incorrectly.
- Logic errors are mistakes made by the programmer.

Suppose we have a program that determines if a person can vote in an election and you must be 18 years or older to vote.

```
if (age > 18) { // This means that age cannot be 18 thus 18 yearolds would not be able to vote. 18 is n
    std::cout << "Yes you can vote" << endl;
}
```

4.8 Section challenge solution

```
#include <iostream>
int main() {
    int favorite_number;
    std::cout << "Enter your favorite number between 1 and 100: ";
    std::cin >> favorite_number;
    std::cout << "Amazing!! Thats my favorite number too!" << std::endl;
    std::cout << "No really!!, " << favorite_number << " is my favorite number!" << std::endl;
}
/* Output:
Enter your favorite number between 1 and 100: 67
Amazing!! Thats my favorite number too!
No really!!, 67 is my favorite number!
*/
```

Chapter 5

Structure of a C++ program