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, Illutstrator, MySQL, MonggoDB.
  - 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.

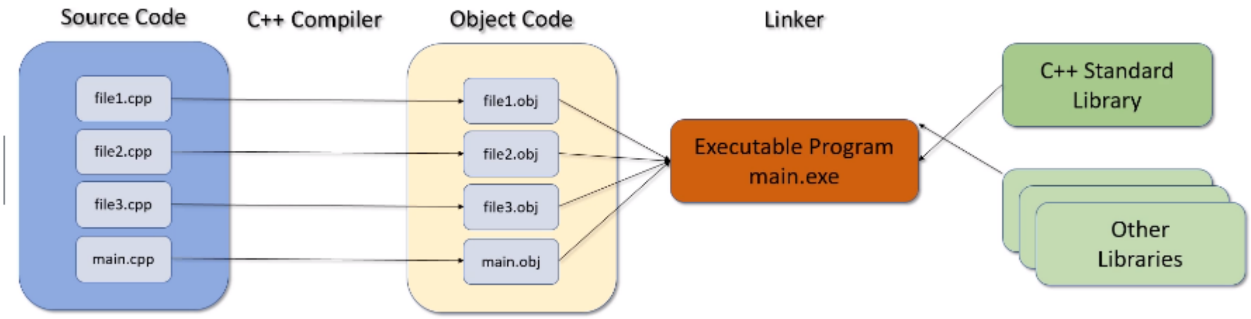
1.2.1 Modern C++ and C++ Standard

- 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.