

COMP 322: Introduction to C++

Chad Zammar, PhD

Apr 09, 2021



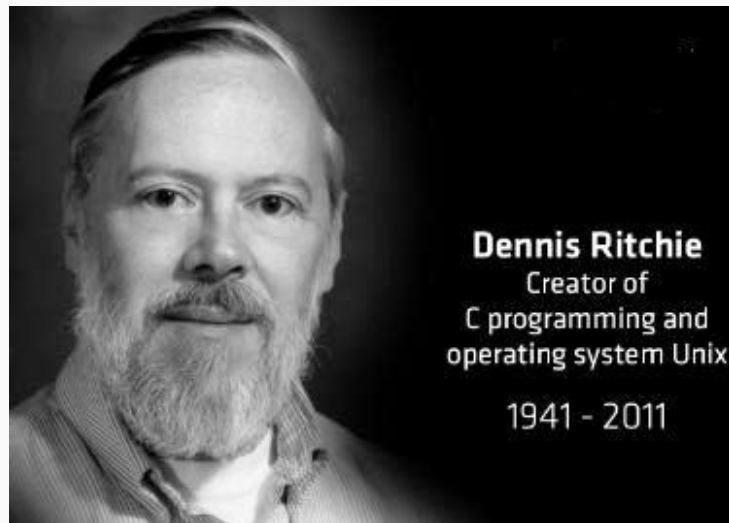
Lecture 10

(Past, present and future)

- The rise of C++
- The “new” releases
- The Future ...

Before the ++

- C++ was born from C, and C was made for system programming
- C is probably the first portable computer programming language to exist
- C is probably one of the oldest programming languages to survive that long and being always actively used in industry
- C was the source of multiple derivative languages other than C++, like objective-C and C#



Before the ++

- C syntax is adopted by many programming languages that were not necessarily derivative of it
 - Java, PHP, Perl ...
- Dennis Richie, from an interview (<http://www.linuxfocus.org/English/July1999/article79.html>):
C++ benefited enormously from C, because C had a fairly large acceptance even before the growth of C++, and C++ could use C both as a base to build a new language and as a tool to create its compilers. C has been characterized (both admiringly and invidiously) as a portable assembly language, and C++ tries to lift its level to object orientation and a more abstract approach to programming.

UNIX is basically a simple operating system, but you have to be a genius to understand the simplicity.

— *Dennis Ritchie* —

The rise of C++

- C++ brought abstraction to C
- Adoption within industry:
 - High Performance Computing, gaming, embedded systems, financial applications and many more.
- The dude to the right is Bjarne Stroustrup, creator of C++



Bjarne Stroustrup, why I created C++

<https://www.youtube.com/watch?v=JBjinqG0BP8>



C++ recent releases

- C++ had a major refactoring with the release of C++11. The previous major release before that was C++98
 - C++03 was a minor release in between
- C++17 is also considered a major release
 - C++14 was a minor release in between
- The home of standard C++:
 - <https://isocpp.org/>

What's new in C++11

- **Support for Concurrency and a Threading library as part of the language**
- **Lambda expressions**
- **Automatic type deduction**
- **More support for initialization**
 - **In-class initialization of class data members**
 - **Container initialization**
- **Introduction of strongly typed nullptr (replacing NULL pointer)**

What's new in C++11

- **Support for few smart pointer classes**
- **Addition of new algorithms**
- **Constructor delegations**
 - **A constructor can call another constructor within the same class**
- **New category of reference type: Rvalue**

What's new in C++17

- **Some features were being deprecated**
 - Remove ++ for bool type
 - Remove auto_ptr since it was replaced by more powerful smart pointers
 - Removing c library headers: <ccomplex>, <cstdbool> ...
 - ...
- **New core language features were being added**
 - Strict order of expression evaluation
 - Exception specification as part of the type system
 - Dynamic allocation for over-aligned types
 - Template <auto>
 - UTF8
 -

What's new in C++17

- **New library features**
 - **Math special functions**
 - **File system**
 - **Parallelism**
 - **More new algorithms**
 - ...
- **Note that since C++17, C++11 is referred to as the official standard**

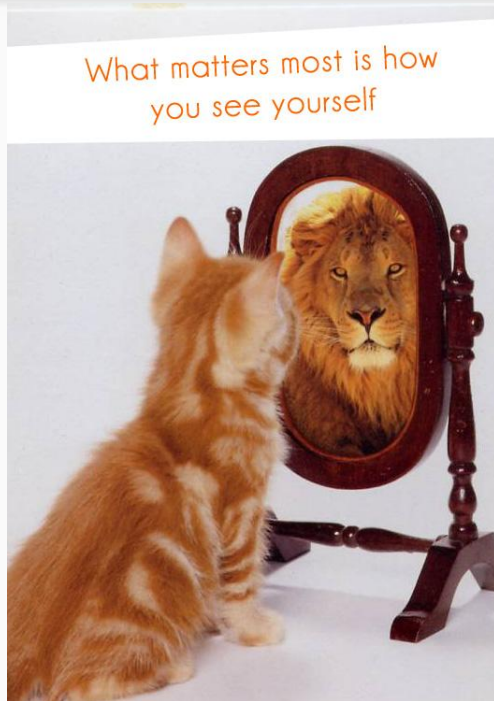
C++20

- C++20 was released in december 2020
- Some features:
 - Reflection
 - Metaclasses
 - Atomic smart pointers
 - Some networking utilities
 - ...

What's still lacking in C++?

- **Standard support for libraries such as XML etc.**
- **Garbage collection**
- **Advanced meta-programming features**
- **Standard support for graphical user interface**

C++: The Lion of Programming Languages



- **To the left: C++ developer after compiling his code and not getting error messages**

Good luck folks!