Modern C++ Programming

1. Introduction

Federico Busato

University of Verona, Dept. of Computer Science
2018, v1.0

About Programming

"And programming computers was so fascinating. You create your own little universe, and then it does what you tell it to do"

Vint Cerf, *TCP/IP co-inventor and Turing Award*

"Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program"

Linus Torvalds, *principal developer of the Linux kernel*

"You might not think that programmers are artists, but programming is an extremely creative profession. It's logic-based creativity"

John Romero, *co-founder of id Software*

A Little History of C and C++ Programming Languages

The Assembly Programming Language

A long time ago, in a galaxy far, far away....there was **Assembly**

- Extremely simple instructions
- Requires lots of code to do simple tasks
- Can express anything your computer can do
- Hard to read, write
- ...redundant, boring programming, bugs proliferation

```
main:
.Lfunc_begin0:
    push rbp
.Lcfi0:
.Lcfi1:
    mov rbp, rsp
.Lcfi2:
    sub rsp, 16
    movabs rdi, .L.str
.Ltmp0:
    mov al, 0
    call printf
    xor ecx, ecx
    mov dword ptr [rbp - 4], eax
    mov eax, ecx
    add rsp, 16
    pop rbp
    ret
.Ltmp1:
.Lfunc_end0:
.L.str:
    .asciz "Hello World\n"
```

In the 1969 **Dennis M. Ritchie** and **Ken Thompson** (AT&T, Bell Labs) worked on developing a operating system for a large computer that could be used by a thousand users. The new operating system was called **UNIX**.

The whole system was still written in assembly code. Besides assembler and Fortran, UNIX also had an interpreter for the **programming language B**. A high-level language like B made it possible to write many pages of code task in just a few lines of code. In this way the code could be produced much faster then in assembly.

A drawback of the B language was that it did not know data-types. (Everything was expressed in machine words). Another functionality that the B language did not provide was the use of "structures". The lag of these things formed the reason for Dennis M. Ritchie to develop the **programming language C**. In 1988 they delivered the final standard definition ANSI C.

Dennis M. Ritchie, and Ken Thompson

```c
#include "stdio.h"

int main() {
    printf("Hello World\n");
}
```

Areas of Application:

- UNIX operating system
- Computer games
- Due to their power and ease of use, C were used in the programming of the special effects for Star Wars

Star Wars - The Empire Strikes Back

The **C++ programming language** (originally named "C with Classes") was devised by **Bjarne Stroustrup** also an employee from Bell Labs (AT&T). Stroustrup started working on C with Classes in 1979. (The $++$ is C language operator)

The first commercial release of the C++ language was in October of 1985

Bjarne Stroustrup

Areas of Application

- *Operating systems:* Windows, Android, OS X, Linux
- *Image Editing Application:* Adobe Premier, Photoshop and Illustrator
- *Web browser:* Firefox, Chrome, etc.
- High-Performance Computing (HPC)
- Embedded systems
- Multimedia
- *Scientific applications:* Machine Learning, Data analysis at CERN/NASA, SETI@home
- Google also use C++ for Indexing
- *Database:* MySQL
- *Compilers:* LLVM
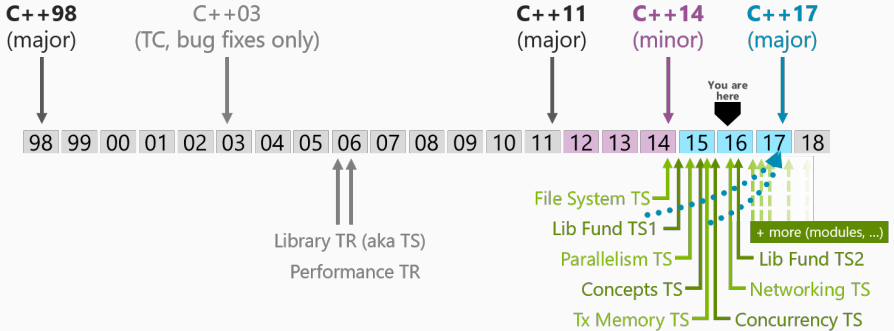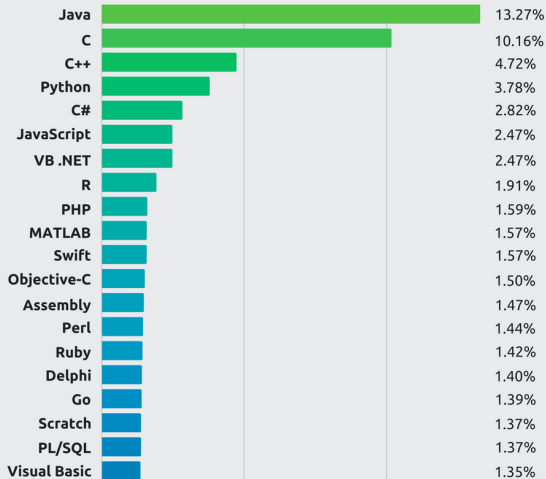
Modern C++ Evolution

C++

C++ Philosophy

- Only add features if they solve an actual problem

- Programmers should be free to choose their own style

- **Compartmentalization** is key

- Allow the programmer **full control** if they want it

- Don't sacrifice **performance** except as a last resort

- Enforce **safety at compile time** whenever possible

Top Programming Languages
Tiobe Index - December 2017

Language	Percentage
Java	13.27%
C	10.16%
C++	4.72%
Python	3.78%
C#	2.82%
JavaScript	2.47%
VB .NET	2.47%
R	1.91%
PHP	1.59%
MATLAB	1.57%
Swift	1.57%
Objective-C	1.50%
Assembly	1.47%
Perl	1.44%
Ruby	1.42%
Delphi	1.40%
Go	1.39%
Scratch	1.37%
PL/SQL	1.37%
Visual Basic	1.35%

Why C++ is so popular?

- **Extreme performance** (theoretically enables highest performance)
- **Relatively easy to prove and test a C++ program**
- **Many support tools:** coverage, analysis, profiling, etc.
- **Low-level code:** drivers, kernels, etc.

N-BODY SIMULATION
PROGRAMMING LANGUAGES PERFORMANCE COMPARISON

Programming Language	Execution Time (S)
C++	8.23
GO	21.47
SWIFT	21.96
JAVA	22.1
Node.js	26.61
PHP	300
Ruby	360
Perl	660
Python3	780
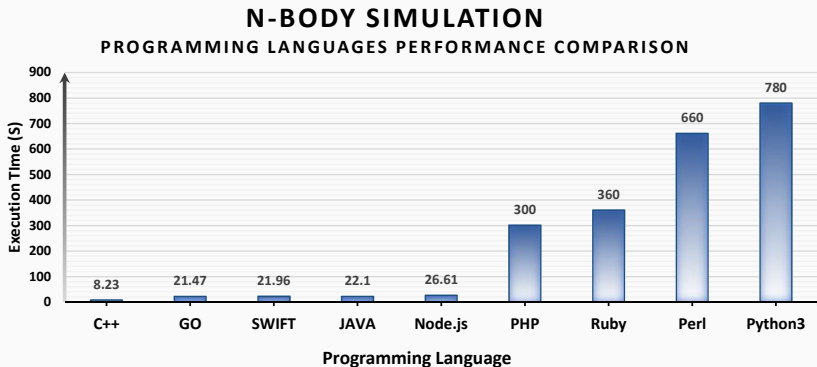
Why C++ is so difficult?

- C++ is the hardest language for students to master
- Huge set of features
- Worry about memory management
- Learn meta-programming
- Distinguish compile-time from run-time
- Low-level implementation details: pointer arithmetics, structure padding, etc.

"C makes it easy to shoot yourself in the foot; C++ makes it harder, but when you do it blows your whole leg off"

Bjarne Stroustrup

"The problem with using C++...is that there's already a strong tendency in the language to require you to know everything before you can do anything"

Larry Wall *(developer of the Perl language)*

References

Unofficial C++ references:

```
en.cppreference.com
```

```
www.cplusplus.com/reference
```

```
IBM Knowledge Center
```

Tutorials:

```
www.learncpp.com
```

```
www.tutorialspoint.com/cplusplus
```

```
en.wikibooks.org/wiki/C++
```

```
yet another insignificant...programming notes
```

Other resources:

```
isocpp.org/faq
```

```
stackoverflow.com
```

The Course

What is/What is not

What the course **is not**:

- A theoretical course on programming
- A high-level concept description

What the course **is**:

- A quite advanced C++ programming language course
- A very practical course
- Don't focus on the concepts behind something but on what is the best way to use it
- Prefer examples instead long descriptions
- Present many language features

Prerequisites:

- Knowledges of C programming language
- Knowledges of object-oriented programming

Who We Are

Federico Busato

- `Research interests:` Parallel/High-Performance Computing and Graph Theory
- `Previous Experience:` Software Engineer at Nvidia (Santa Clara, California, USA)
- `Other Courses:` Advanced Architectures (Master degree), Operating System (Bachelor degree)

Alessandro Danese

- `Research interests:` Embedded System Verification
- `Previous Experience:` Software Engineer at Intel (Portland, Oregon, USA)
- `Other Courses:` Design Automation of Embedded Systems (Master degree), Operating System (Bachelor degree)

"The only way to learn a new programming language is by writing programs in it"

Dennis Ritchie