UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

INTRODUCTION TO MODERN C++

Lecture 1

Rémi Géraud

January 28, 2016

École Normale Supérieure de Paris

Prelude

This course: overview

- Fast-paced, practice-oriented introduction to *modern* C++

THIS COURSE: OVERVIEW

- Fast-paced, practice-oriented introduction to *modern* C++
- No previous programming experience needed (but it helps)

This course: overview

- Fast-paced, practice-oriented introduction to *modern* C++
- No previous programming experience needed (but it helps)
- Requires personal work between lectures

This course: lectures, labs, homework

- Each lecture will cover a key aspect of C++ programming.

This course: lectures, labs, homework

- Each lecture will cover a key aspect of C++ programming.
- Each lab will confront you with real-life problems.

This course: lectures, labs, homework

- Each lecture will cover a key aspect of C++ programming.
- Each lab will confront you with real-life problems.
- There will be some homework for you. It's optional.

This course: lectures, labs, homework

- Each lecture will cover a key aspect of C++ programming.
- Each lab will confront you with real-life problems.
- There will be some homework for you. It's optional.
- I took examples from video games, finance, and maths.

THIS COURSE: SYLLABUS AND EXAM

- We will not cover every aspect of C++
- But we will discuss essential aspects as we progress
- The exam will depend on that progress

Contact and general info

- Contact me:
  `remi.geraud@ens.fr`
- Textbook:
  B. Stroustrup, *Programming — Principles and Practice Using C++*
- 12 Lectures:
  30% Theory, 70% Practice

Lecture 1

History, legacy, and dangers of C++

TABLE OF CONTENTS

WHAT IS C++?

What is C++? And why do we care?

- Short answer:

  A popular and influential (old) programming language.

- Long, uninformative answer:

  Statically typed, free-form, multi-paradigm, compiled, general-purpose, intermediate-level, ISO/IEC 14882:2014.

Almost every program you've used was made with some C++.

IN A NUTSHELL

People usually like C++ because

- C++ is almost as **fast** as you can get
- C++ gives you near-perfect **control** of the device
- C++ is **expressive** enough to describe most things
- C++ is **portable** from one machine to another
- C++ looks familiar and **seems readable**.

Back in 1979, this was a revolution (same year as Pacman).

“MODERN” C++

- In this course, we will learn the most recent standard, C++14
- We will use constructs absent from legacy C++
- We don’t code in 2016 like they did in 1979.

Programme (for now)

Lec. 1 Course presentation. History, legacy, and dangers of C++.
TD: “Hello world”, the compilation process, toolchain.

Lec. 2 Types, Declarations, Statements. Arithmetic.
TD: Basic maths, Arrays, Vectors, and the Quake 3 trick.
Homework: Black-Scholes option pricing

Lec. 3 Computing with C++: Control flow, Functions, STL.
TD: Sorting, Complexity, Numerical integration, and Angry birds.

Lec. 4 Debugging, Testing, Proving.
TD: Exceptions, Assertions, and the Apple TLS bug.

Lec. 5 Pointers, References, and Lambdas in C++.
TD: Memory allocation, Linked lists, File I/O, Overflows.
Homework: Monte-Carlo integration

Lec. 6 Classes, Namespaces, RAII.
TD: Symbolic computation 1

WHAT IS A PROGRAMMING LANGUAGE?

PROGRAMMING LANGUAGE I

- Computers are **stupid**: They need clear, simple instructions. You can't just ask "What's my most important e-mail today?"
- They are good at simple, repetitive tasks: $2 + 2$
- But it is often **boring and risky** to do micromanagement

"Low-level" programming language

e.g. asking an intern to make coffee

- For many years, low-level was the only thing.
- It looks like this:

```
lea     0x24ae41(%rip),%rax
test    %rax,%rax
je      4e4056
mov     (%rax),%rdx
xor     %esi,%esi
jmpq    42b740
xor     %edx,%edx
jmp     4e404f
```

- Error prone, boring — what does it do?

Programming language III

- We can hire **translators** that break down “complex” orders into a sequence of elementary operations
- You can then focus only on **higher-level tasks**:
  “Compute 2^5” $\Rightarrow$ “$2 \times 2 \times 2 \times 2 \times 2$”
- This makes life easier, if you can trust the translator.

“Compiled” programming language

We will look at compilation in the lab.

Programming language IV

Summary:

- Write commands in a complex, powerful language
- Translate (=compile) them into simple, trivial tasks

Now we just have to learn the language (C++) and hire the translator.

C++ is indeed a **language**, with

- **Syntax**, i.e. correct order of tokens in a statement;
- **Grammar**, i.e. a way to attribute a role to tokens;
- **Semantics**, i.e. meaning of a statement;
- **Style**, i.e. a preferred way to state things;
- **Dialects**, **Users**, **Grammar nazis**, etc.

Important note: C++ wasn't designed, it was grown.

PROGRAMMING LANGUAGE VI

That’s probably how C++ looks to you now

Μῆνιν ἄειδε, θεά, Πηληϊάδεω Ἀχιλῆος
οὐλομένην, ἣ μυρί’ Ἀχαιοῖς ἄλγε’ ἔθηκε,
πολλὰς δ’ ἰφθίμους ψυχὰς Ἄϊδι προΐαψεν
ἡρώων, αὐτοὺς δὲ ἑλώρια τεῦχε κύνεσσιν
οἰωνοῖσί τε δαῖτα· Διὸς δ’ ἐτελείετο βουλή·
ἐξ ου δὴ τὰ πρῶτα διαστήτην ἐρίσαντε
Ἀτρεΐδης τε ἄναξ ἀνδρῶν καὶ δῖος Ἀχιλλεύς.

In a few weeks you will be able to **read**, **write** and **explain**.

HISTORY AND LEGACY OF C++

HISTORY OF C++

You don’t study the history of English before learning English.

Same here.

LEGACY OF C++

Much more interesting is the **impact** that C++ has had.

- Influenced the design of many programming languages: Rust, Go, Java, C#, PHP; Javascript, Objective-C...
- Brought OOPs to the programming community
- Still dominating the industry (this is changing)
- Attracted much criticism for its complexity and fragility

Most new languages try to **fix** the flaws while **keeping the goods**.

What's wrong with C++?

Why would I tell you it’s dangerous?

Why would I tell you it's dangerous? Because it is.

THE DANGERS OF C++

Why would I tell you it's dangerous? Because it is.

- The programmer is in charge of *everything*

THE DANGERS OF C++

Why would I tell you it's dangerous? Because it is.

- The programmer is in charge of *everything*
- That includes security,

THE DANGERS OF C++

Why would I tell you it's dangerous? Because it is.

- The programmer is in charge of *everything*
- That includes security, correctness,

THE DANGERS OF C++

Why would I tell you it's dangerous? Because it is.

- The programmer is in charge of *everything*
- That includes security, correctness, memory,

THE DANGERS OF C++

Why would I tell you it's dangerous? Because it is.

- The programmer is in charge of *everything*
- That includes security, correctness, memory, files, etc.

THE DANGERS OF C++

Why would I tell you it's dangerous? Because it is.

- The programmer is in charge of *everything*
- That includes security, correctness, memory, files, etc.
- There is no safety net.

THE DANGERS OF C++

Why would I tell you it's dangerous? Because it is.

- The programmer is in charge of *everything*
- That includes security, correctness, memory, files, etc.
- There is no safety net. This is a loaded weapon.

THE DANGERS OF C++

Why would I tell you it's dangerous? Because it is.

- The programmer is in charge of *everything*
- That includes security, correctness, memory, files, etc.
- There is no safety net. This is a loaded weapon.

"With great power comes great responsibility"

THE DANGERS OF C++

Why would I tell you it's dangerous? Because it is.

- The programmer is in charge of *everything*
- That includes security, correctness, memory, files, etc.
- There is no safety net. This is a loaded weapon.

"With great power comes great responsibility"
"Power corrupts ; absolute power corrupts absolutely"

SETUP AND TOOLS

What do we need to write C++?

It was not clear whether you'd bring your own laptops. **Please do**.

You need:

1. An OS (I **strongly** recommend Ubuntu/VirtualBox)
2. A text editor (`vim` or `gedit`)
3. A compiler (`clang++` or `g++`)

On Ubuntu you can install everything by typing

```
sudo apt-get update
sudo apt-get install vim gedit clang++ g++
```

That's all you need. Install that!

Questions?

Lab Session
"Hello world" and the compilation process