Université de Paris

Programmation avancée
1 – Introduction to C++

Sylvain Lobry

19/09/2022

Today’s menu

1 History

2 C++ 101

History

A short history of C++

- 1979: Bjarne Stroustrup finishes PhD on distribution computer systems at Cambridge. Leaves for Bell Labs. Start working on ”**C with classes**”

Original motivation: needed to write code to work on distributed computers:

  - Simula: makes it easy, but performances issues
  - C: good performance but not modular enough

- 1984: renames it to C++
- 1985: first commercialization
- → 1998: first standard: core language and standard library
- 2011: important update, C++11 (followed by C++14, C++17, C++20)

History

A short history of C++

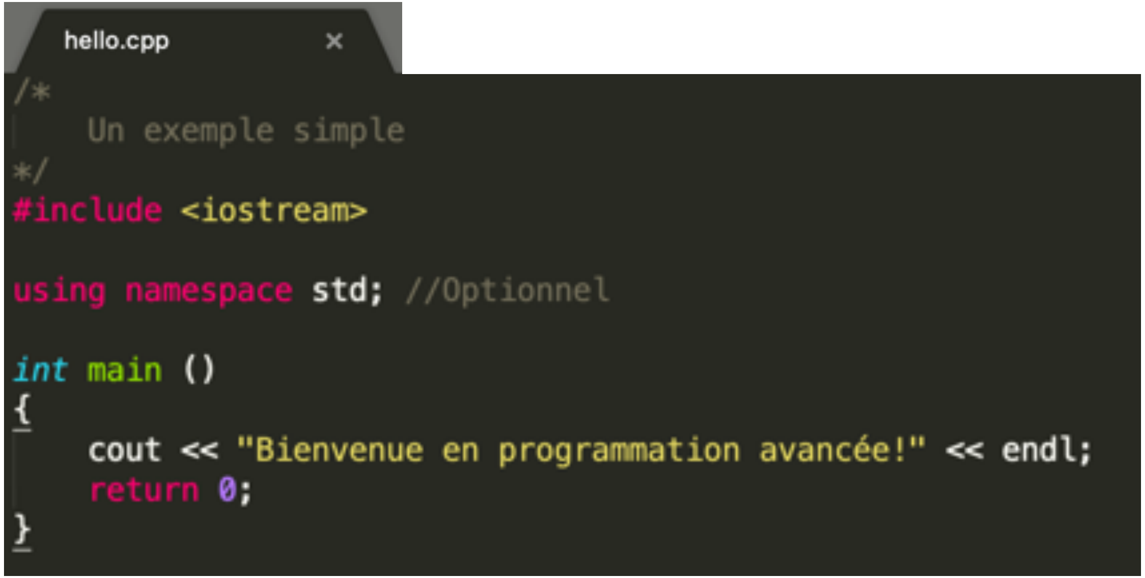
- It is an extension of C (hence the name)
- Most valid C programs are also valid C++ programs
- Allows for high-level abstractions (but also low-level programming and a lot of freedom) -> best of both worlds?
- Complex language...

> *“The problem that I have with them today is that... C++ is too complicated.”*
>
> Donald Knuth

- Today, we assume that you know C

C++ 101

What does a C++ program look like?



```
/*
    Un exemple simple
*/
#include <iostream>

using namespace std; //Optionnel

int main ()
{
    cout << "Bienvenue en programmation avancée!" << endl;
    return 0;
}
```

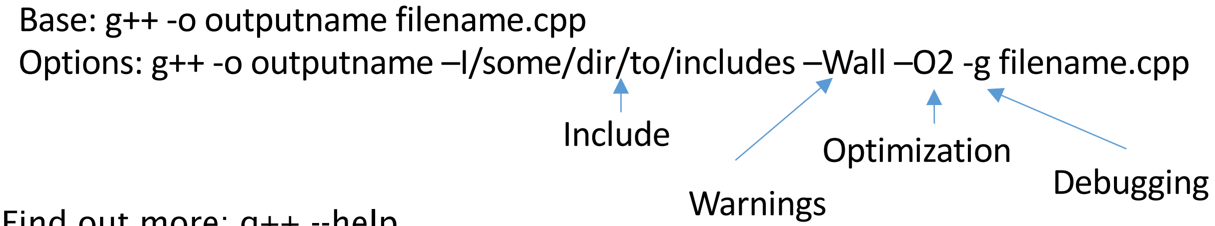
Which similarities with C?

C++ 101

Compile C++ code

- Installing a compiler, most common choice
  - On Unix: g++
  - On Windows: Visual C++
- A compiler implements the C++ standard
- Command line (on Unix):

⚠ Some might implement new features faster, undefined behaviors are… undefined.



```
Base: g++ -o outputname filename.cpp
Options: g++ -o outputname –I/some/dir/to/includes –Wall –O2 -g filename.cpp
```

- Find out more: g++ --help

C++ 101

Inputs/Outputs

- As most of the time, you *can* use standard C I/O procedures.. But:
- Streams: abstraction to perform I/O
- Defined in the header iostream of the standard library (namespace: std)
- Standard output stream: cout
  - Use the insertion operator << to insert data
  - Can be chained
  - Can be used with different types of variables
  - To add a new line: ‘\n’ or endl

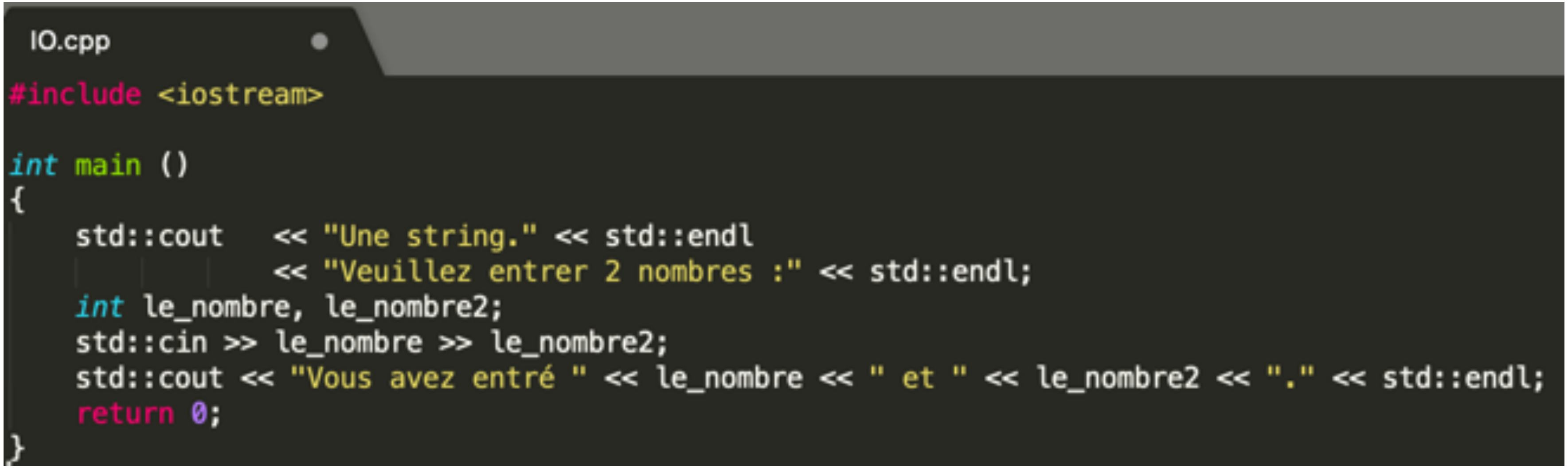
C++ 101

Inputs/Outputs

- As most of the time, you *can* use standard C I/O procedures.. But:
- Streams: abstraction to perform I/O
- Defined in the header iostream of the standard library (namespace: std)
- Standard output stream: cout
    - Use the insertion operator << to insert data
    - Can be chained
    - Can be used with different types of variables
    - To add a new line: ‘\n’ or endl
- Standard input stream: cin
    - Use the insertion operator **>>** to insert data
    - Can also be chained
    - Can be used with different types of variables
- Also: clog, cerr

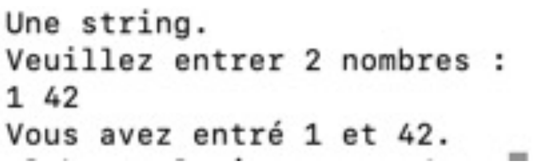
C++ 101

Inputs/Outputs



```cpp
#include <iostream>

int main ()
{
    std::cout   << "Une string." << std::endl
                << "Veuillez entrer 2 nombres :" << std::endl;
    int le_nombre, le_nombre2;
    std::cin >> le_nombre >> le_nombre2;
    std::cout << "Vous avez entré " << le_nombre << " et " << le_nombre2 << "." << std::endl;
    return 0;
}
```

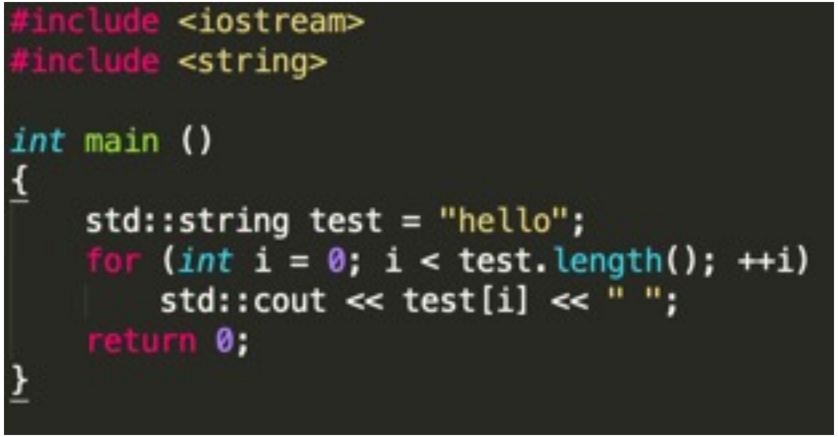


```
Une string.
Veuillez entrer 2 nombres :
1 42
Vous avez entré 1 et 42.
```

C++ 101

Strings

- Defined in header… string
- 2 ways to initialize (as for other variables):
    - std::string test = ‘’hello’’;



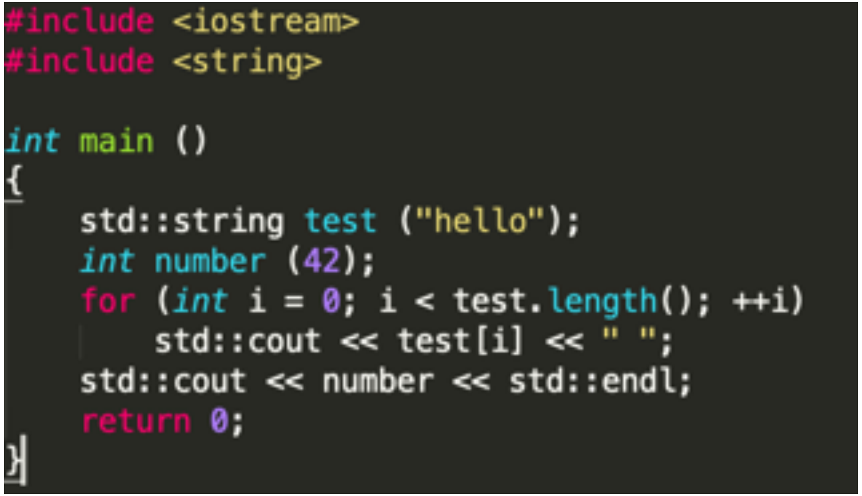
```
#include <iostream>
#include <string>

int main ()
{
    std::string test = "hello";
    for (int i = 0; i < test.length(); ++i)
        std::cout << test[i] << " ";
    return 0;
}
```

C++ 101

Strings

- Defined in header… string
- 2 ways to initialize (as for other variables):
    - std::string test = ’’hello’’;
    - std::string test (‘’hello’’);



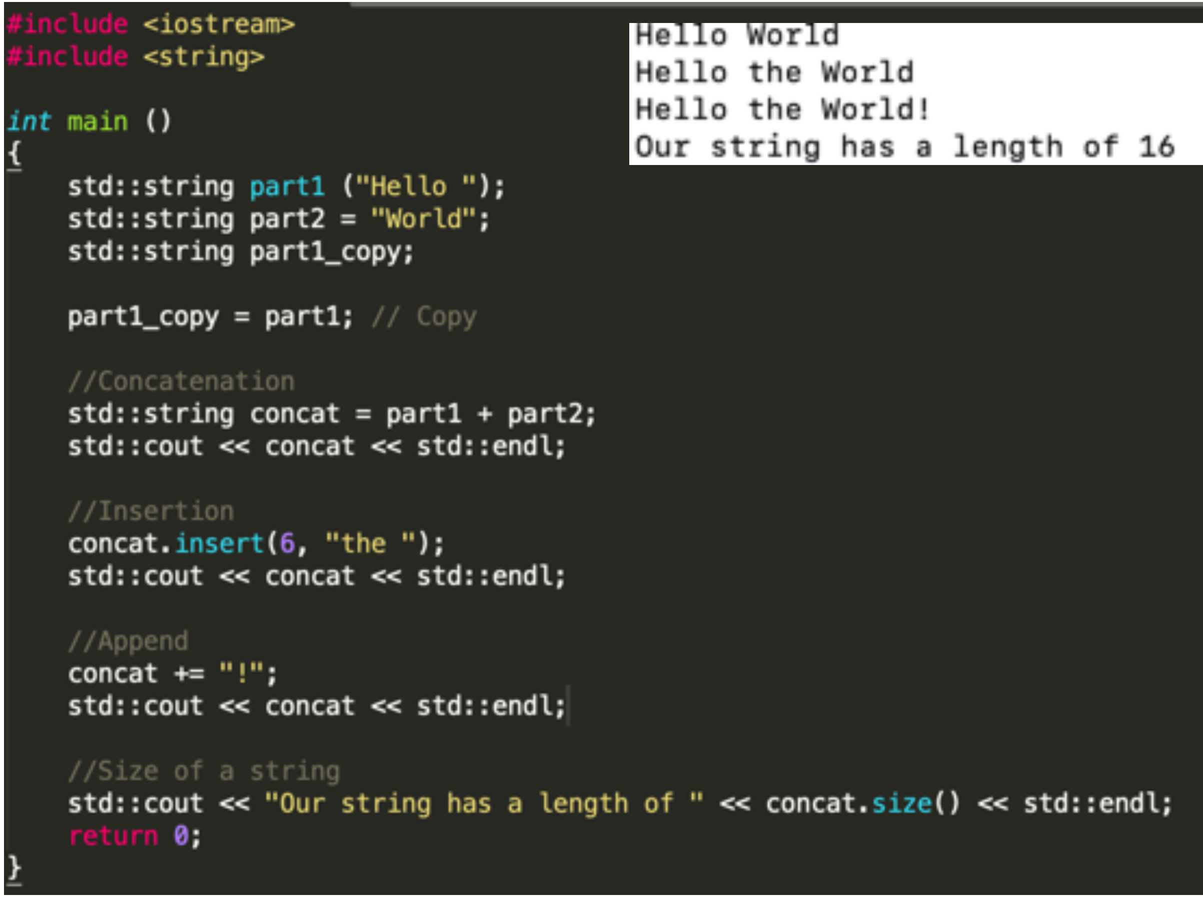
```
#include <iostream>
#include <string>

int main ()
{
    std::string test ("hello");
    int number (42);
    for (int i = 0; i < test.length(); ++i)
        std::cout << test[i] << " ";
    std::cout << number << std::endl;
    return 0;
}
```

C++ 101

Strings

- Defined in header… string
- 2 ways to initialize:
  - std::string test = ‘’hello’’;
  - std::string test (‘’hello’’);
- Multiple methods defined:
  - Concatenation (operator +)
  - Append (operator +=)
  - Insertion (method insert)
  - Size (method size)
  - …



```
#include <iostream>
#include <string>

int main ()
{
    std::string part1 ("Hello ");
    std::string part2 = "World";
    std::string part1_copy;

    part1_copy = part1; // Copy

    //Concatenation
    std::string concat = part1 + part2;
    std::cout << concat << std::endl;

    //Insertion
    concat.insert(6, "the ");
    std::cout << concat << std::endl;

    //Append
    concat += "!";
    std::cout << concat << std::endl;

    //Size of a string
    std::cout << "Our string has a length of " << concat.size() << std::endl;
    return 0;
}
```

```
Hello World
Hello the World
Hello the World!
Our string has a length of 16
```

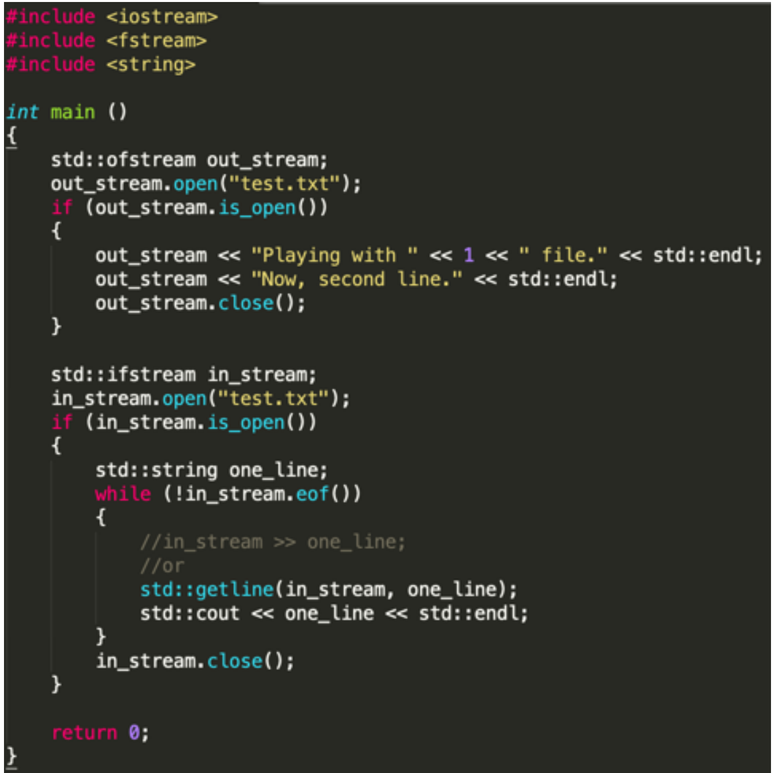
C++ 101

Inputs/Outputs… with files

- Similar syntax.
- Streams defined in header fstream
- Open:
  - Using an object of class fstream: fstream my_file; my_file.open(filename, mode)
  - mode can be (among others)
    - ios::in | ios::out | ios::binary
  - ifstream and ofstream classes: different default modes
  - Can be checked using my_file.is_open();
- Close: my_file.close();
- Write: <<
- Read: >> or std::getline() (for strings)

C++ 101

Inputs/Outputs… with files



```cpp
#include <iostream>
#include <fstream>
#include <string>

int main ()
{
    std::ofstream out_stream;
    out_stream.open("test.txt");
    if (out_stream.is_open())
    {
        out_stream << "Playing with " << 1 << " file." << std::endl;
        out_stream << "Now, second line." << std::endl;
        out_stream.close();
    }

    std::ifstream in_stream;
    in_stream.open("test.txt");
    if (in_stream.is_open())
    {
        std::string one_line;
        while (!in_stream.eof())
        {
            //in_stream >> one_line;
            //or
            std::getline(in_stream, one_line);
            std::cout << one_line << std::endl;
        }
        in_stream.close();
    }

    return 0;
}
```

C++ 101

Questions?

- https://www.wooclap.com/PROGAC1