Lec 1: Introduction & Logistics

Bernard Nongpoh

CS 1214: Object Oriented Programming and Data Structures Lab

L	T	P	C
2	0	3	7

Tuesday: **10 AM to 10:55 AM**
Wednesday: **11 AM to 11:55 AM**
Friday: Quizzes/Compensation Class
Classroom: **5401**

Monday: 2 PM to 5 PM
Lab: CCC

Website: https://pass-iitg.github.io/cs1214/

CS 1214: Assessment Policy

Component	Weight (%)
Quizzes	**10%**
Assignments	**25%**
Labs	**25%**
Mid-Semester Exam	**15%**
End-Semester Exam	**25%**

Github Classroom

To ensure smooth onboarding and accurate grading, please follow the instructions below carefully:

- Create a **GitHub account using your official IIT Guwahati (IITG) email ID**.
- **Grading will be done strictly based on the official IITG email ID** linked to your GitHub account.
- To map your GitHub username with your official details, **please submit the required information using the form below**.

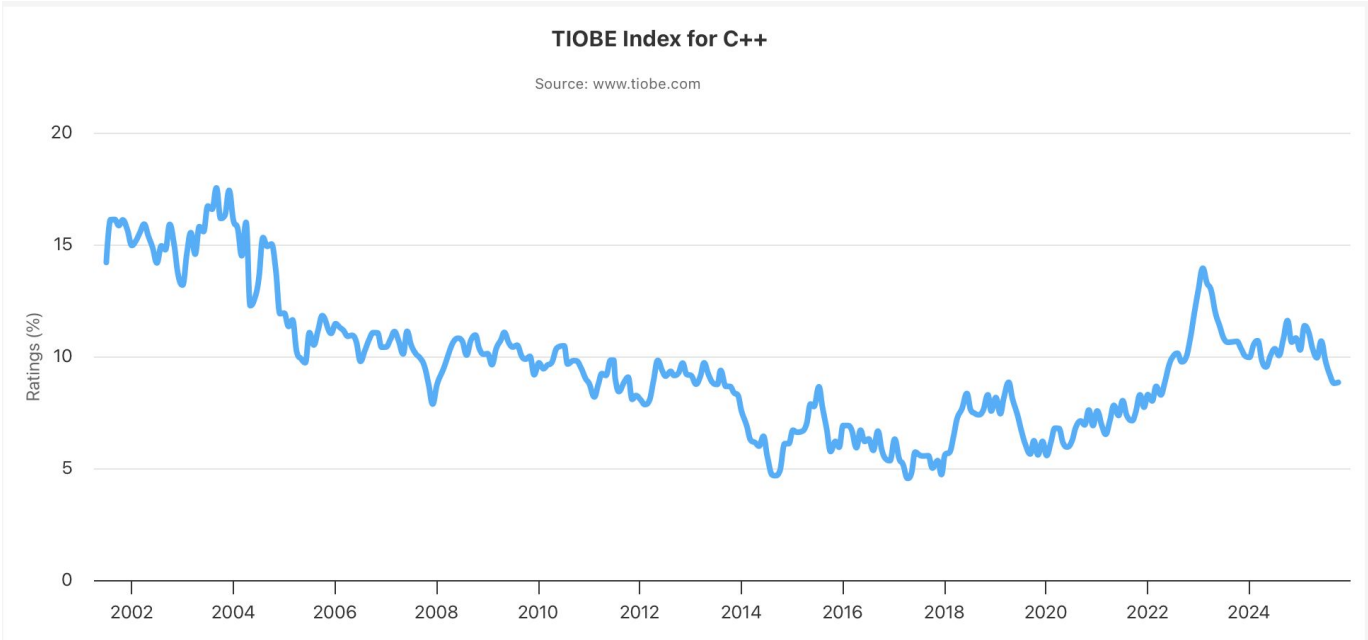
Google Form:
https://forms.gle/QLNyxJ2hDenauANP6

Failure to complete this step may result in difficulties with assignment access or grading.

Why Learn C++

- One of the world’s most widely used programming languages
- Use to build games, browsers, operating systems, and AI systems

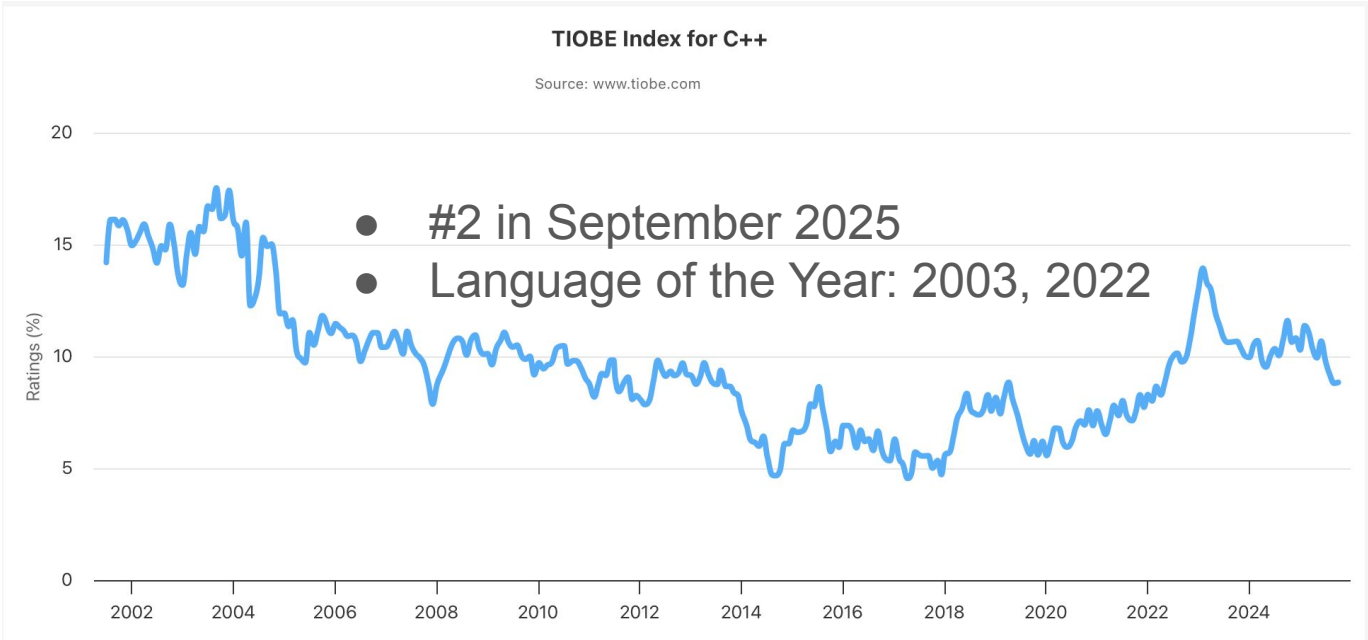
Why Learn C++



Growth trajectory

https://www.tiobe.com/tiobe-index/cplusplus/

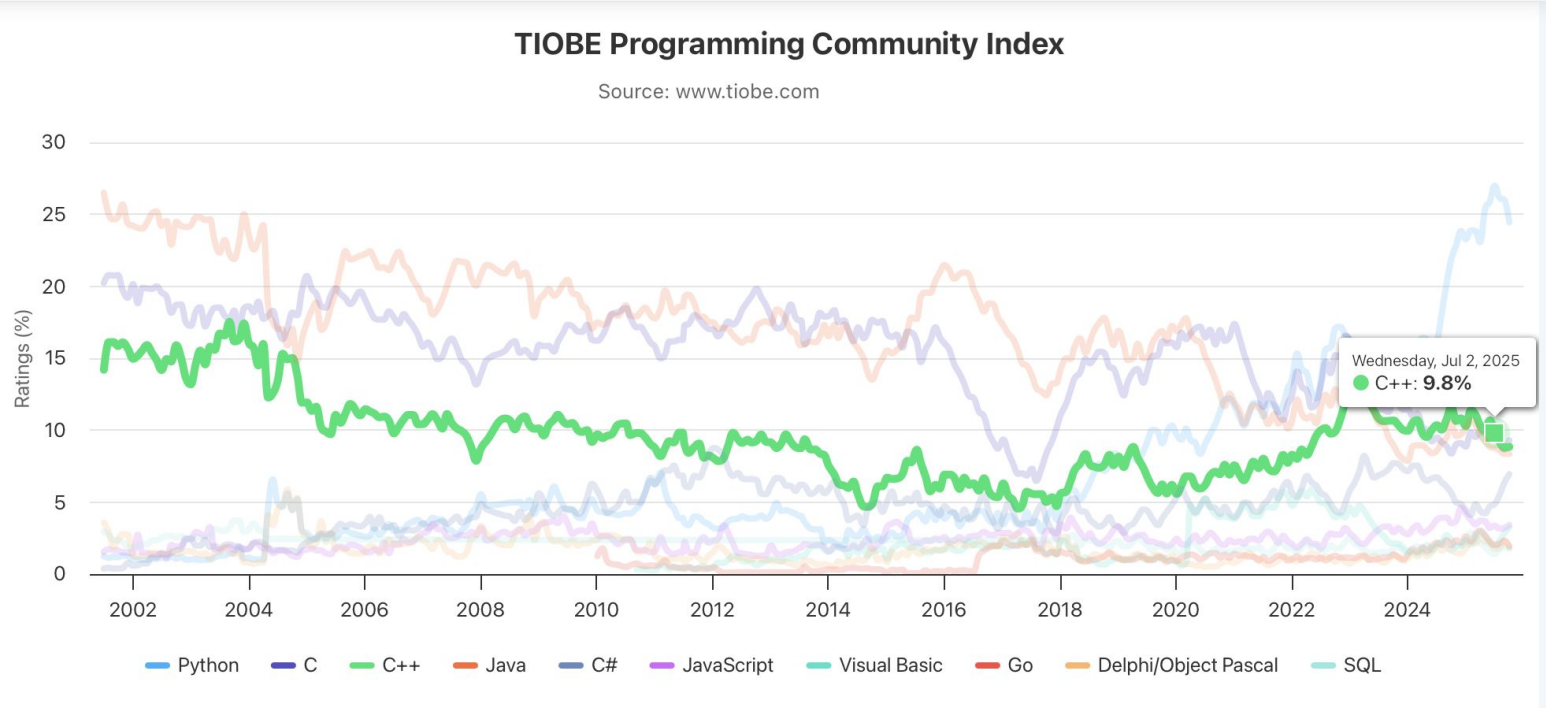
Why Learn C++



Growth trajectory

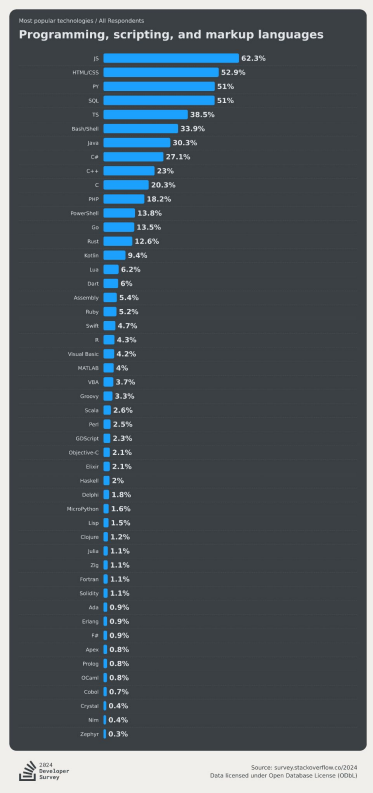
https://www.tiobe.com/tiobe-index/cplusplus/

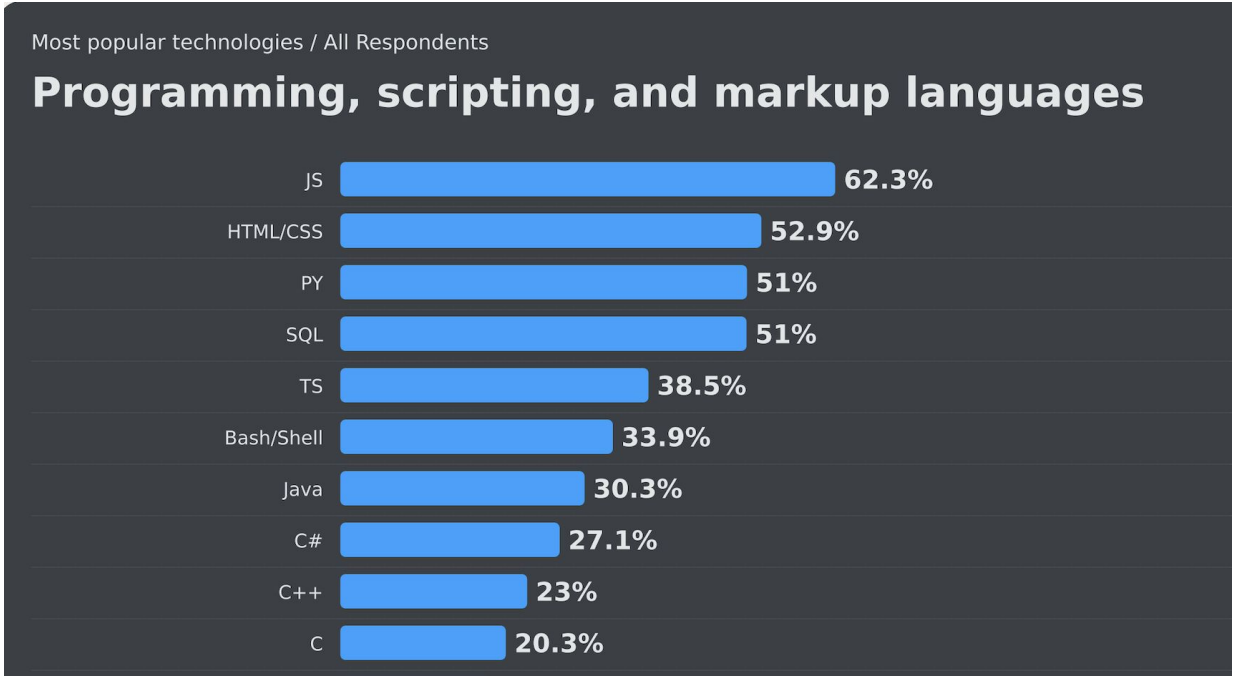
Why Learn C++



https://www.tiobe.com/tiobe-index/cplusplus/

Why Learn C++

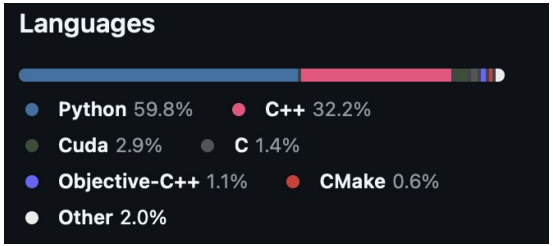




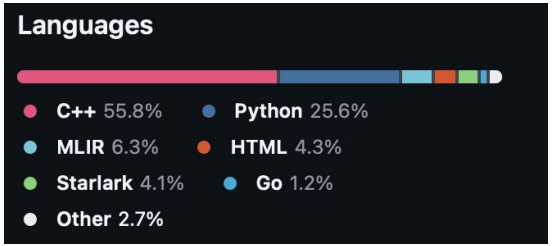
https://survey.stackoverflow.co/2024/technology#most-popular-technologies-language

Why Learn C++

PyTorch



TensorFlow



Why Learn C++

Source

Why Learn C++

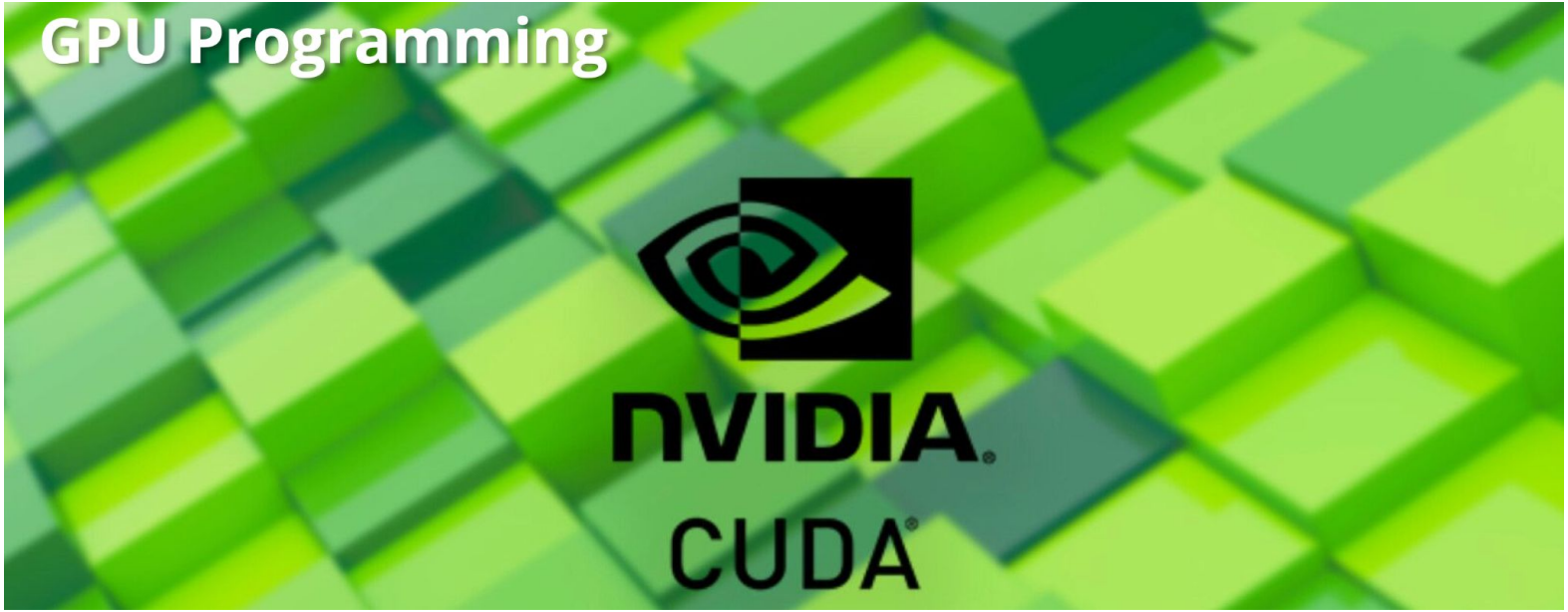
Source

Why Learn C++

High Frequency Trading

Source

Why Learn C++



Source

History

- Bjarne Stroustrup

  Danish computer scientist

  known for the development of the C++ programming language.

https://en.wikipedia.org/wiki/Bjarne_Stroustrup

History

- Simula is the initial source of C++’s abstraction mechanisms. The class concept (with derived classes and virtual functions) was borrowed from it
- However, templates and exceptions came to C++ later with different sources of inspiration
- 1979 under the name “C with Classes”
- 1984: C with Classes was renamed to C++
- 1985: C++ was released commercially

Basic C++ Programming Elements

HelloWorld.cpp

```cpp
/*
 * Hello, World in C++
*/
#include<iostream>
int main()
{
  std::cout<< "Hello, World!\n"; // Using iostream
  return 0; // terminate successfully
}
```

Basic C++ Programming Elements: Comments

HelloWorld.cpp

```
1 /*
2  * Hello, World in C++
3 */
4 #include<iostream>
5 int main()
6 {
7   std::cout<< "Hello, World!\n"; // Using iostream
8   return 0; // terminate successfully
9 }
```

- Multi-line comment starts with `/*` and ends with `*/`
- Ignored by the compiler
- Used to describe what the program does
- For single line, `// comment`

Basic C++ Programming Elements: Preprocessor Directive

HelloWorld.cpp

```
1 /*
2  * Hello, World in C++
3 */
4 #include<iostream>
5 int main()
6 {
7   std::cout<< "Hello, World!\n"; // Using iostream
8   return 0; // terminate successfully
9 }
```

- **`#include`** tells the **preprocessor** to include another file before compilation
- **`<iostream>`** provides access to **standard input/output streams** (`cin`, `cout`, `cerr`, `clog`)
- It's part of the **C++ Standard Library**
-

Basic C++ Programming Elements: Preprocessor Directive

HelloWorld.cpp

```
1 /*
2  * Hello, World in C++
3 */
4 #include<iostream>
5 int main()
6 {
7   std::cout<< "Hello, World!\n"; // Using iostream
8   return 0; // terminate successfully
9 }
```

- **`#include`** tells the **preprocessor** to include another file before compilation
- **`<iostream>`** provides access to **standard input/output streams (`cin`, `cout`, `cerr`, `clog`)**
- It's part of the **C++ Standard Library**
-

We're importing a library that lets us talk to the screen and keyboard

Basic C++ Programming Elements: **The `main()` Function**

HelloWorld.cpp

```
1 /*
2  * Hello, World in C++
3 */
4 #include<iostream>
5 int main()
6 {
7   std::cout<< "Hello, World!\n"; // Using iostream
8   return 0; // terminate successfully
9 }
```

- Every C++ program **must have a `main()`** - it's the **entry point**
- **`int`:** the function returns an integer to the OS
- **`{ ... }`**: body of the function (statements executed in order)

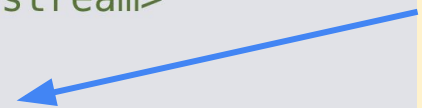
When **`main()`** finishes, the OS checks its return value:

- **`0`** : success
- non-zero: error or abnormal termination

Basic C++ Programming Elements: **Output Statement**

HelloWorld.cpp

```
1 /*
2  * Hello, World in C++
3 */
4 #include<iostream>
5 int main()
6 {
7   std::cout<< "Hello, World!\
8   return 0; // terminate succ
9 }
```

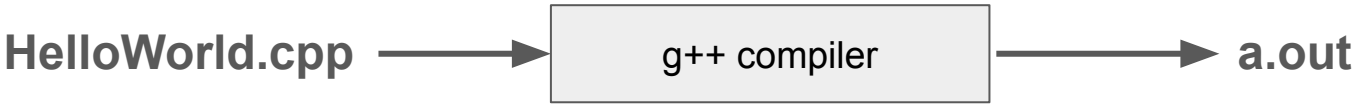


- **std:** standard namespace (contains C++ library features)
- **cout**: standard output stream (prints to screen)
- **std::cout**: use **cout** from the **std** namespace
- **<<**: stream insertion operator, sends data to output
- **"Hello, World!"** is a string literal to display
- **\n** is a newline character (moves cursor to next line)
- **Needs <iostream>**: defines **std::cout**
- **Type-safe:** automatically handles different data types
-

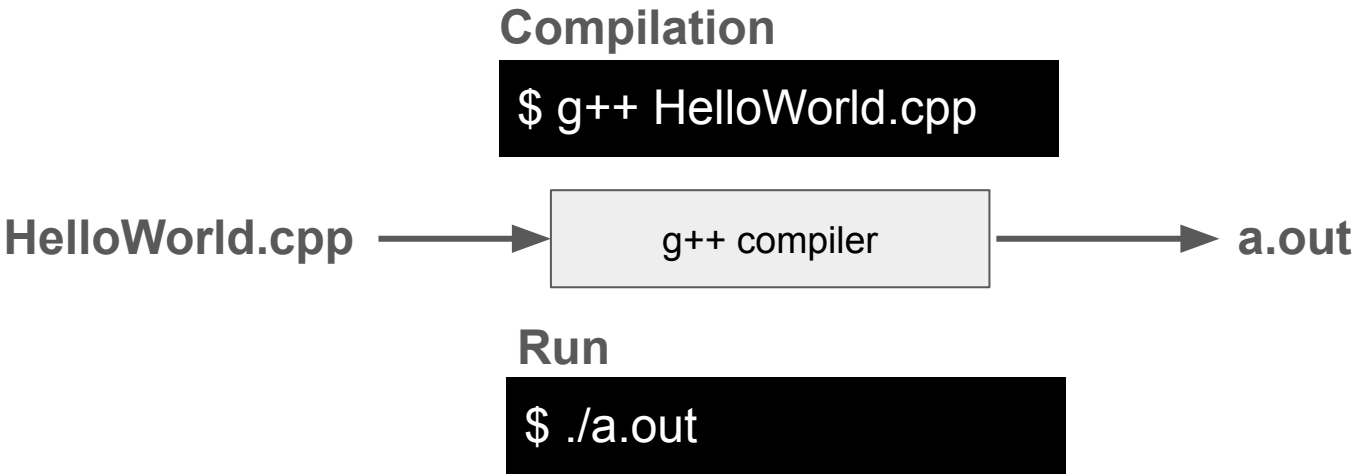
Basic C++ Programming Elements: **Compilation Process**

Compilation

```
$ g++ HelloWorld.cpp
```



Basic C++ Programming Elements: **Compilation Process**



Basic C++ Programming Elements: **Compilation Process**

Compilation

```
$ g++ HelloWorld.cpp -o HelloWorld
```

Basic C++ Programming Elements: **Compilation Process**

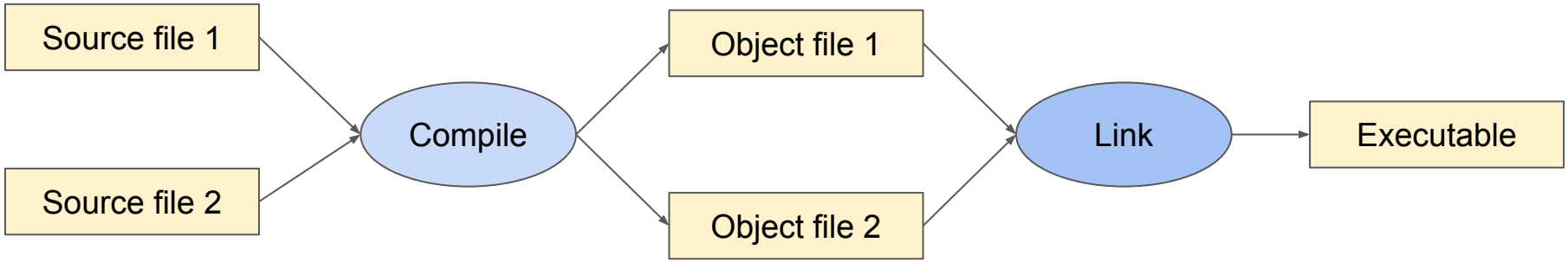
Compilation

```
$ g++ HelloWorld.cpp -o HelloWorld
```

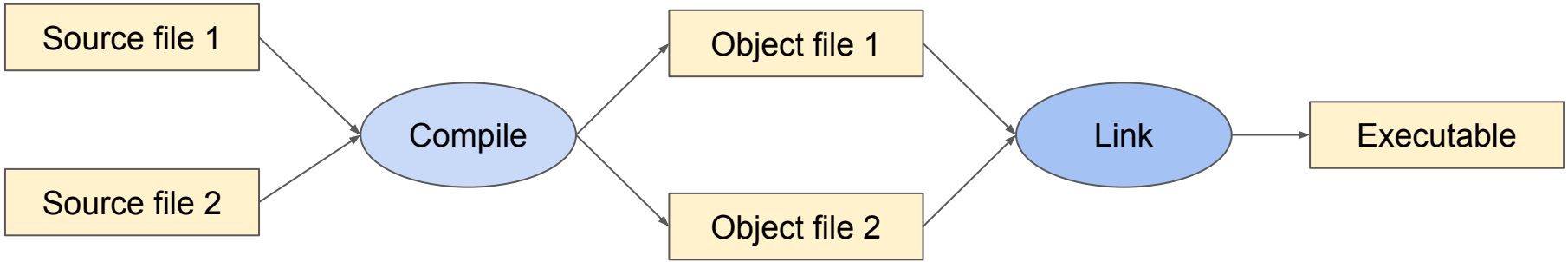
Run

```
$ ./HelloWorld
```

Basic C++ Programming Elements: **Compilation Process**



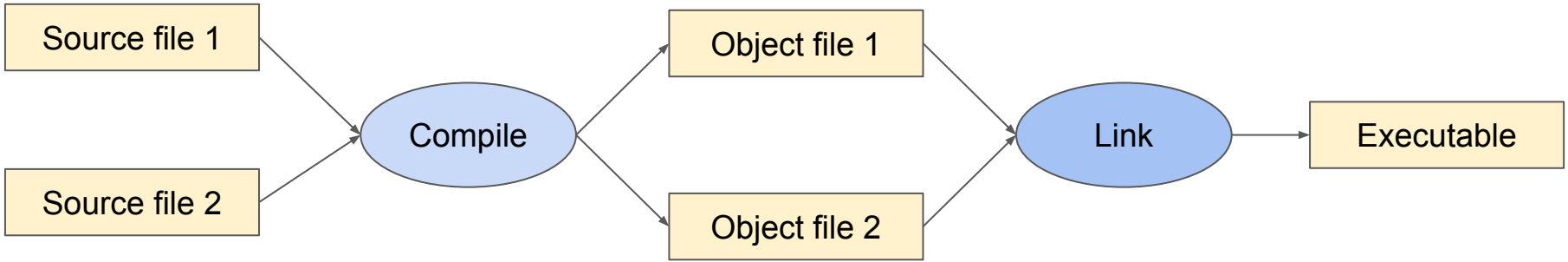
Basic C++ Programming Elements: **Compilation Process**



C++ is a **compiled language**

- For a program to run, it source text has to be processed by a compiler, producing object files, which are combined by a linker yielding an executable program

Basic C++ Programming Elements: **Compilation Process**



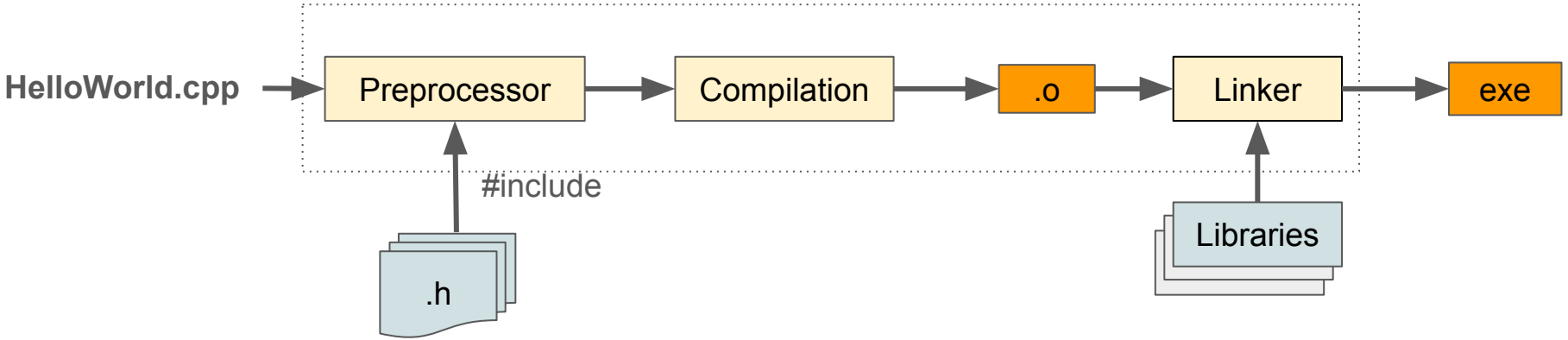
C++ is a **compiled language**

- For a program to run, it source text has to be processed by a compiler, producing object files, which are combined by a linker yielding an executable program

C++ is a **statically type language**

- The type of every entity(eg object, value, name, and expression) must be know to the compiler at its point of use

Basic C++ Programming Elements: **Compilation Process**



Basic C++ Programming Elements

hello_world.cpp

```cpp
/*
 * Hello, World in C++
*/
#include<iostream>
int main()
{
  std::cout<< "Hello, World!\n"; // Using iostream
  return 0; // terminate successfully
}
```

C/C++ Compilers

Clang/LLVM

GNU Compiler Collection

Microsoft Visual C++

GodBolt Compiler Explorer

https://godbolt.org/