C++ Program Design -- Introduction

Junjie Cao @ DLUT

Summer 2021

https://github.com/jjcao-school/c

What the C++ language is and what we can use it for?

Different languages for different purposes

- What is computer programming?

- Natural language vs. programming language
- What natural language is?
  - **a language is a tool for expressing and recording human thoughts**.
  - use language for speaking, writing, reading, listening and **thinking**
  - allowing us all to understand and to be understood
  - accompanies us throughout our entire lives

Different languages for different purposes

- What natural language is?
- What programming language is?
  - Communication between **Human & Machine**, Human & Human
  - defined by a certain set of rigid rules, much more inflexible than any natural language. these rules determine:
    - **Lexicon词汇:** which symbols符号 (letters, digits, punctuation marks标点, and so on) could be used
    - **Syntax句法:** how to compose a valid program
    - **Semantics语义:** what syntactically valid programs mean

    - Any program must be error-free in these three ways: lexically, syntactically and semantically, otherwise, the program won't run.
    - It is sure that you'll encounter all of these errors, as to err is human and it's these fallible humans who write the computer programs.

Different languages for different purposes

- Machine language vs. high-level programming language

- A computer, is like a well-trained dog – it responds only to a predetermined **set of known commands**.
  - “take that number, add to another and save the result”.
  - an example x86 machine language instruction: 10110000 01100001

- A complete set of well-known commands is called an **instruction list (IL, 指令集)**.
- The IL is known as **machine language:** the computer’s mother tongue.
- Computer programming: composing instructions to cause a desired effect.

Different languages for different purposes

- Machine language vs. high-level programming language

- The IL is known as **machine language:** the computer's mother tongue.
  - In early stages, it was the only available method of programming: 10110000 01100001
  - This kind of programming is tedious, time-consuming and **highly susceptible to a programmer's mistakes**.
  - ILs of different types of computers may be entirely different.

  - => high-level programming language: **bridge** between the people's language (natural language) & computer language (machine language)
    - The translation is done by a computer, making the whole process fast and efficient.
    - This makes machine language: **Machine & Machine.**

Different languages for different purposes

- Machine language vs. high-level programming language

- High-level programming language
  - **bridge** between the people's language (natural language) & computer language (machine language)

  - The **translation** is done by a computer, making the whole process fast and efficient.

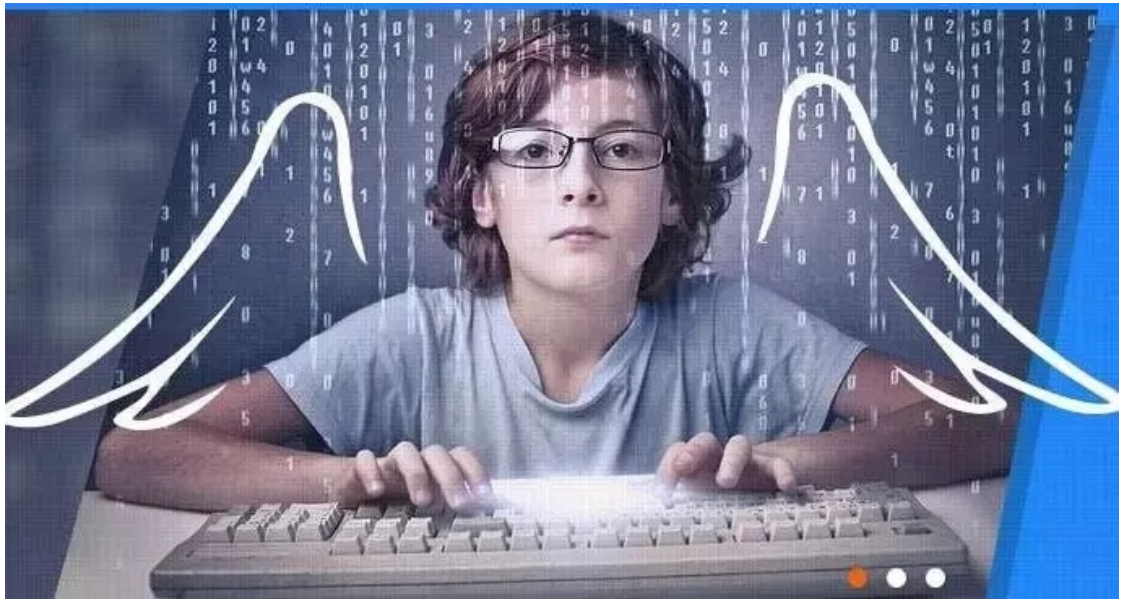
  - **Portability可移植性**
    - the programs written in high-level languages can be translated into any number of different machine languages, => enable them to be used on many different computers

Translation via a Compiler

- The translation is made by a specialized computer program called a **compiler 编译器**.
  - Write the program in accordance with the rules of the chosen programming language.
  - This program (which in fact is just a text) is called the **source code**, or simply source, while the file that contains the source is called the **source file (cpp, cc**, **cp**, **cxx**, **c++)**.
  - Compilation => an **executable file (exe，可执行程序)** composed of machine language

  - Of course the whole process is actually a bit more complicated.
    - Linking by linker链接器
    - We will say these later.

Coding is important



- 万物有眼（传感器），万物有脑（程序），万物互联
- 许多工作岗位逐步被机器所替代或融合
- 幼儿编程；Alpha Go；你呢？

- When human beings acquired language, we learned not just how to listen but how to speak.
- When we gained literacy, we learned not just how to read but how to write.
- And as we move into an increasingly digital reality, we must **learn not just how to use programs but to make them**.

- In the emerging, highly programmed landscape ahead, you will either create the software or you will be the software. It's really that simple: Program, or be programmed.

- Choose the former, and you gain access to the control panel of civilization. Choose the latter, and it could be the last real choice you get to make.

-- Douglas Rushkoff, in Program or Be Programmed: Ten Commands for a Digital Age (2010)

本学期的目标

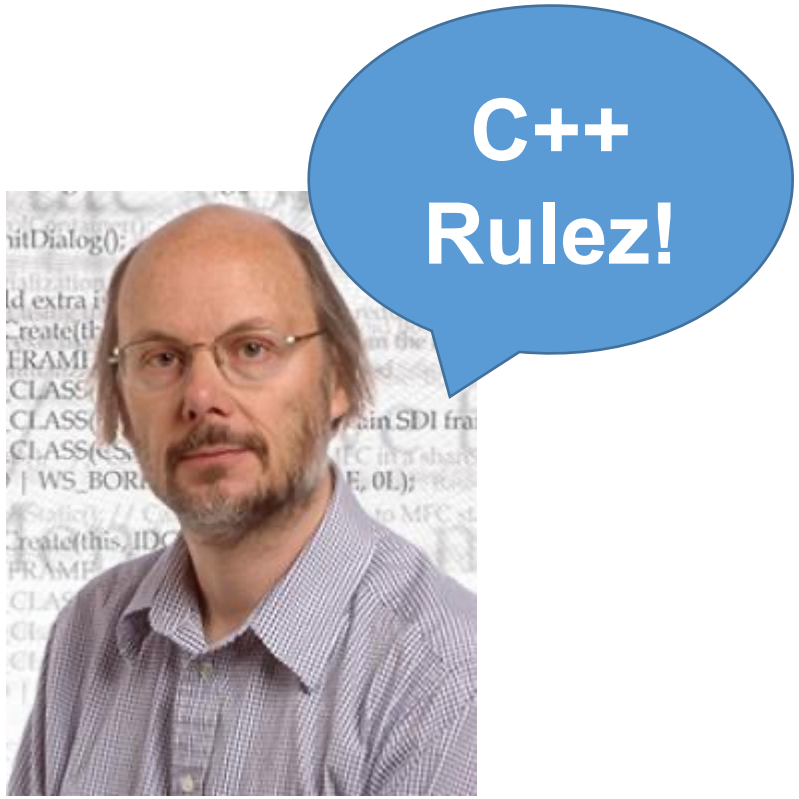
给定数据+问题描述，独立写程序解决这个问题

What the C++ language is and what we can use it for?

Is Matlab/Python the final weapon?

Why teaching C++

Dennis Ritchie
1969 -- 1973 at Bell Labs
C89, …, C99, C11, C18



Bjarne Stroustrup: Why I Created C++ - YouTube
[bijani sdʒəusdʒup]
1979--1983 at Bell Labs
C++11, C++14, C++17, C++20

Functions, Performance && Maintainable

Functions, Performance && Maintainable

- Functions
  - C is a subset of C++
  - C++是一种混合语言，是集过程化设计、面向对象、基于对象和泛型算法等多种技术于一体的编程语言

- **Performance**
  - Approximately equal to C (<=)
  - In practice, use them instead of any other programming languages, Matlab, Python, Java, etc.

- **Maintainable**
  - **>> C**
  - There are many comparable, even better, languages for large systems
  - But C++ has most of libraries for scientific computing, >> than all other languages

Language evolution

- **Machine Language**
  - The **very limited set** of instructions that a CPU natively understands is called **machine code** (or **machine language** or an **instruction set**)
  - each instruction is composed of a number of binary digits, each of which can only be a 0 or a 1. These binary numbers are often called **bits** (short for binary digit)
    - an example x86 machine language instruction: 10110000 01100001
  - each set of binary digits is **translated** by the CPU into an **instruction** that tells it to do a very specific job
    - compare these two numbers
    - put this number in that memory location.
  - Different types of CPUs will typically have **different** instruction sets, so instructions that would run on a Pentium 4 would not run on a Macintosh PowerPC based computer.
  - Back when computers were first invented, programmers had to **write programs directly in machine language**, which was a very difficult and time consuming thing to do.
- **Assembly Language**
- **High-level Languages**

Language evolution

- **Machine Language**
  - an example x86 machine language instruction: 10110000 01100001
- **Assembly Language汇编语言**
  - each instruction is identified by a **short name** (rather than a set of bits), and **variables** can be identified by names rather than numbers
  - must be translated into machine language by using an **assembler**.
  - Assembly languages tend to be very **fast**, and assembly is still used today when speed is critical.
  - However, the reason assembly language is so fast is because assembly language is tailored to a particular CPU. Assembly programs written for one CPU will **not run on another CPU**.
  - Furthermore, assembly languages still require a lot of instructions to do even simple tasks, and are **not very human readable**.
    - the same instruction as above in assembly language: mov al, 061h
- **High-level Languages**

Language evolution

- Machine Language
  - an example x86 machine language instruction: 10110000 01100001
- Assembly Language
  - the same instruction as above in assembly language: mov al, 061h
- High-level Languages
  - C++: more abstract, easy:
    - Conciseness: 1 = many
    - Maintainability: easier to modify
    - Portability: suitable for different types of processor
  - C++ is a **high-level** language, **compiled** language, strong **types**, **case sensitive**.

```
int main(){
    return 0;
}
```

可维护，更利于人机交流是演化的方向；同时，现实要求有足够的性能

编程语言和思想

- Assembly language
- Computation: Fortran 1954
- System programming: C 1969, **C++** 1979, C# 1999, Objective-C
- Application: Java 1995, Java script, **PHP**
- Unix shell to everything: Perl, **Python**, Ruby
- Computation: **Matlab**, Mathematics, Mapple, R
- **The "concept“ of "programming languages“ are quite "similar"**

Language is the dress of thought.

~Samuel Johnson

But if thought corrupts language, language can also corrupt thought.

~George Orwell

index | TIOBE - The Software

tiobe.com/tiobe-index/

Apps tool research 汉典 WeChat for Web cg daily titan data coding ai dgp

Jun 2021	Jun 2020	Change	Programming Language	Ratings	Change
1	1		C	12.54%	-4.65%
2	3	^	Python	11.84%	+3.48%
3	2	v	Java	11.54%	-4.56%
4	4		C++	7.36%	+1.41%
5	5		C#	4.33%	-0.40%
6	6		Visual Basic	4.01%	-0.68%
7	7		JavaScript	2.33%	+0.06%
8	8		PHP	2.21%	-0.05%

What is the most common use of C++?

- What is C++?
  - An object-oriented 面向对象 programming language
  - C++是一种混合语言，是集过程化设计、面向对象、基于对象和泛型算法等多种技术于一体的编程语言

- Its usage:
  - Large, complex applications
  - Scientific Computing
    - The most of **libraries** for science computation are still implemented in C++.
  - Performance
    - **game**
    - **HPC（高性能计算）**
    - **AI 人工智能底层**
    - **Financial 金融:** 高频交易平台
    - **Operation System, such as windows, Linux**
    - …

Why teaching C++

- Versatile
  - Python >= C++ > Matlab
- 易于掌握
  - Python (free) > Matlab (commercial)
- 性能
  - C++。是Matlab和Pythong的必要补充。
  - Prerequisites
    - **Proficiency in Python, high-level familiarity in C/C++**
      - All class assignments will be in Python, but some of the deep learning libraries we may look at later in the class are written in C++.
      - If you have a lot of programming experience but in a different language (e.g. C/C++/Matlab/Javascript) you will probably be fine.

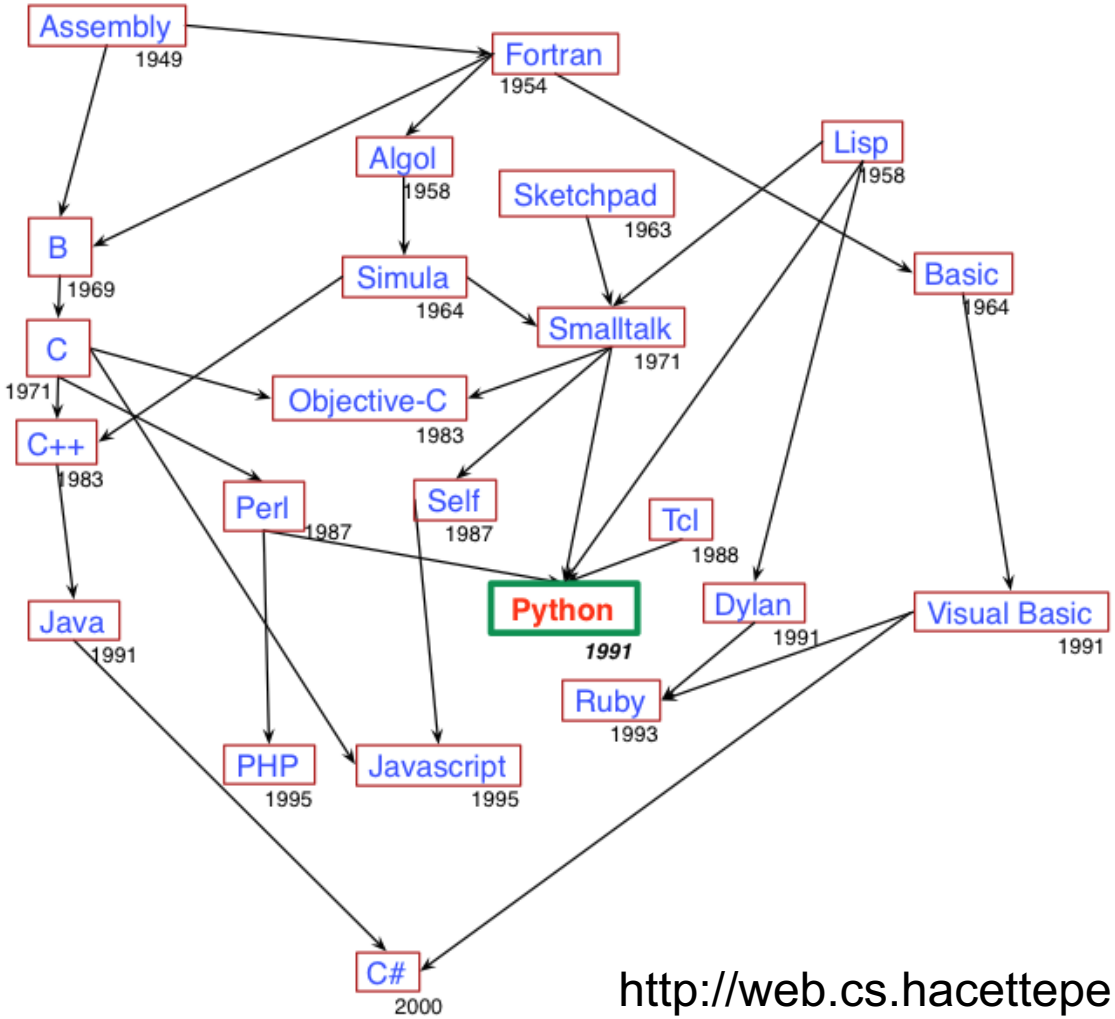
Why teaching C++

- Versatile
  - Python >= C++ > Matlab
- 易于掌握
  - Python (free) > Matlab (commercial)
- 性能
  - C++。是Matlab和Pythong的必要补充。

- Java, Matlab & Python 不适合学习数据结构和算法
- 其它语言不够 **hard**, C++可以用来区分great programmers and mediocre programmers.

Evolution of Programming Languages

The conceptual apparatus used by the C++ language is common for all object programming languages.



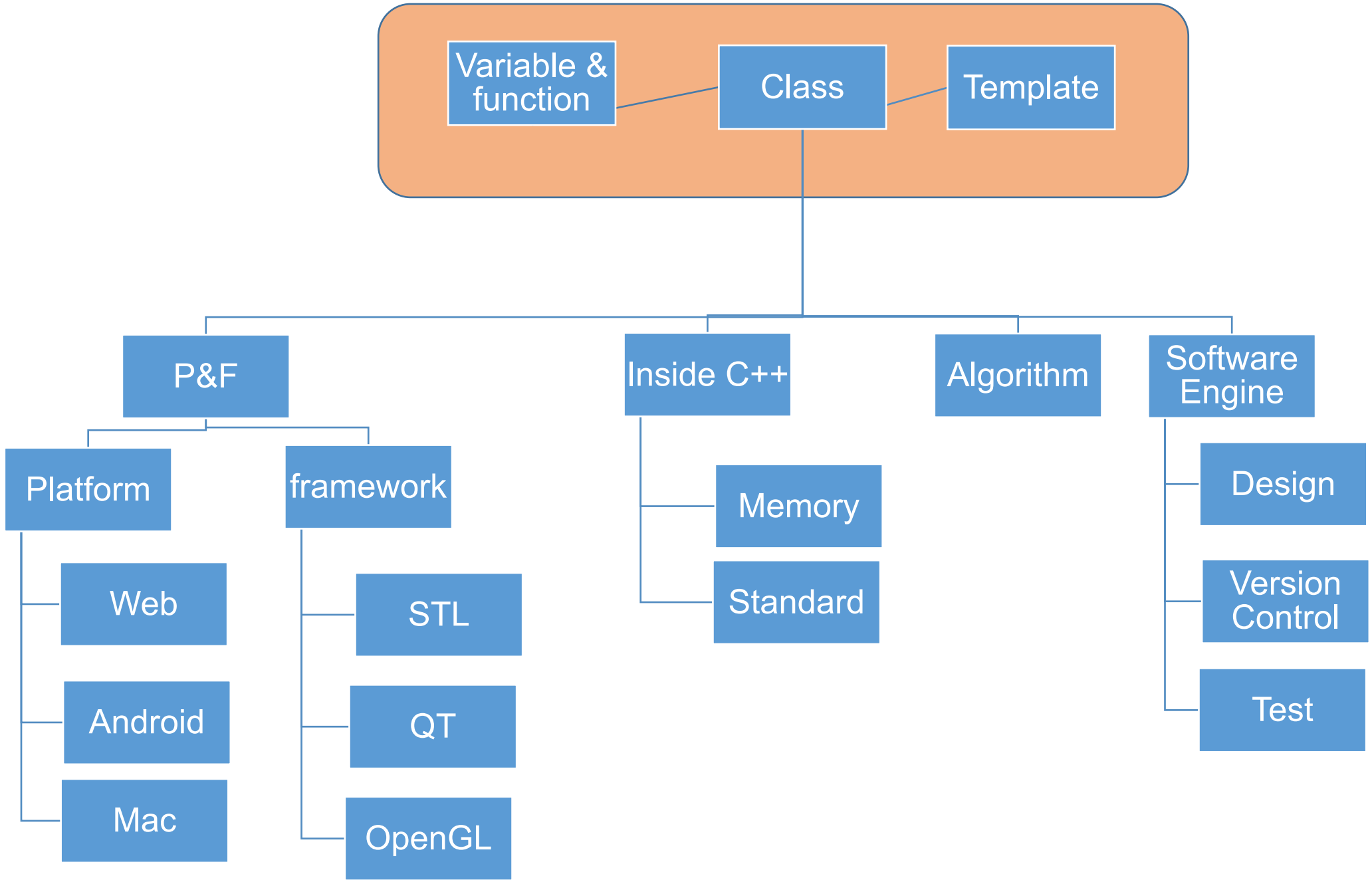
http://web.cs.hacettepe.edu.tr/~bbm101

- 其实这么多年我看着各种库的起起落落，还有一种感慨是研究者**不能始终抱着一个大腿**，要与时俱进。但是时代的潮流在哪里也不是随时都能看出来的，也没法时刻保持自己在前沿，但好在**掌握了一个库之后再换另一个库并不是很费劲**。
  - --CMU LTI博士研究生王赟

课程相关

- 教师
  - Junjie Cao, http://jjcao.github.io
  - jjcao@dlut.edu.cn
- 助教

- 主页: https://github.com/jjcao-school/c


Variable & function
Class
Template
P&F
Inside C++
Algorithm
Software Engine
Platform
framework
Web
Android
Mac
STL
QT
OpenGL
Memory
Standard
Design
Version Control
Test

考核

item	ratio
签到、日常测试、作业	30%
Exam	70%

上机

- 西部校区机房
  - 2-4周的周一下午
  - 1-3周的周四晚
- 做什么
  - 书后习题
  - 后面References中的小题目
  - 4个homework

How to Succeed?

- 56 hours (32 talks + 24 practices) in 4 weeks
- 每个人都可以
  - Work hard
  - 精英日课2: 正确的学习方法只有一种风格
  - 多做编程练习胜过多看书
  - “少想多做”，落实到IDE内；
  - 增量开发，确保每一步可运行:
    - void main() first
    - Function 1
    - Function 2
    - …
  - Debug your code
  - 英文搜索错误信息, Google!!!
  - Learn by good example: follow open source projects
  - **代码行数 约等于 编程能力**

对数学的学生而言;
进阶要求

Video

- The birth of the computer, George Dyson
- SageMath – Open source is ready to compete with Mathematica for use in the classroom, William Stein

程序员 vs 程序猿

控制台程序(Console programs)

远比图形接口程序容易实现和迁移到不同的操作系统

Hello World

```
// A Hello World program
# include <iostream>
int main()
{
    std::cout << "Hello, world!\n";
    return 0;
}
```

Line-By-Line Explanation

- // 注释comment

indicates that everything following it until the end of the line is a **comment**: it is ignored by the compiler.

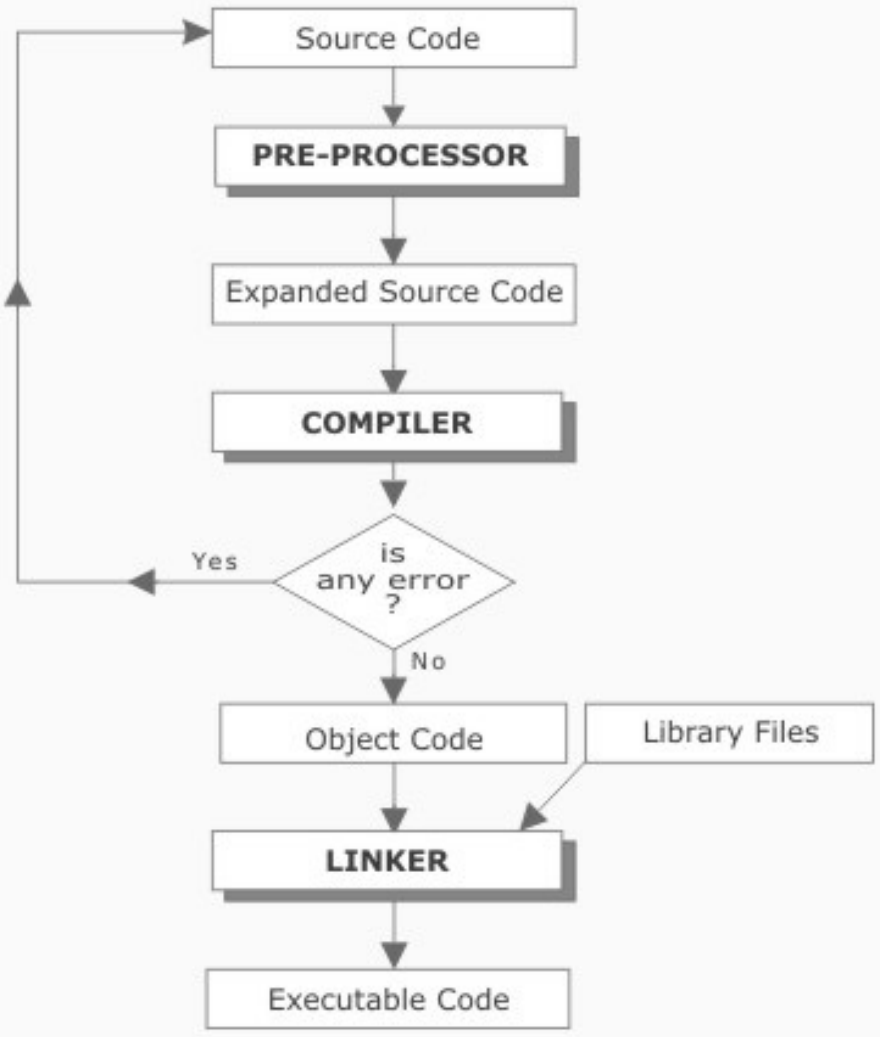
- /* and */
  - (e.g. x = 1 + /*sneaky comment here*/ 1;
  - multiple lines;

```
// A Hello World program
# include <iostream>
int main() {
    std::cout << "Hello, world!\n";
    return 0;
}
```

- Usages
  - Comments exist to **explain non-obvious** things going on in the code. Use them: **document** your code well!

```
// A Hello World program
# include <iostream>
int main() {
    std::cout << "Hello, world!\n";
    return 0;
}
```

- **# preprocessor commands**
  - 用#开始的行是预处理命令(preprocessor commands), which usually change what code is actually being compiled.
  - #include tells the **preprocessor** to dump in the contents of another file, here the iostream file, which defines the procedures for input/output.



Compilation and Linking of a C++ Program

```
// A Hello World program
# include <iostream>
int main() {
    std::cout << "Hello, world!\n";
    return 0;
}
```

- int main()
  - main 函数名
  - 跟随mian的()说明它是一个函数
  - main()之前的int表明该函数返回一个整数值
  - 当程序被执行（载入内存），main()是第一个被执行的函数（程序的入口）

```
// A Hello World program
# include <iostream>
int main() {
    std::cout << "Hello, world!\n";
    return 0;
}
```

- 大括号{}表明main()的函数体
  - {}把多个命令组成一组命令：multiple commands =》a block代码块
  - 每一个命令/声明（command/statement）必须分号结尾
  - More about this syntax in the next few lectures.

```
// A Hello World program
# include <iostream>
int main() {
    std::cout << "Hello, world!\n";
    return 0;
}
```

- cout <<
- This is the syntax for outputting some piece of text to the screen.

```
// A Hello World program
# include <iostream>
int main() {
    std::cout << "Hello, world!\n";
    return 0;
}
```

- std是一个名称空间Namespaces
  - 作用域解析操作符scope resolution operator ::
  - 通知编译器要调用std中的cout，而不是别处jjcao::cout

using namespace std;

- This line tells the compiler that it should look in the std namespace for any identifier we haven’t defined.
- If we do this, we can omit the std:: prefix when writing cout.

```
// A Hello World program
# include <iostream>
int main() {
    std::cout << "Hello, world!\n";
    return 0;
}
```

- 字符串String
  - *Hello, world*
  - 像这样显示指定的字符串，叫string literal.字符串字面量

- \n
  - The \n indicates a **newline** character.
  - 转义序列（Escape sequences）: It is an example of an escape sequence – a symbol used to represent a special character in a text literal.

```
// A Hello World program
# include <iostream>
int main() {
    std::cout << "Hello, world!\n";
    return 0;
    std::cout << "Hello, again!\n";
}
```

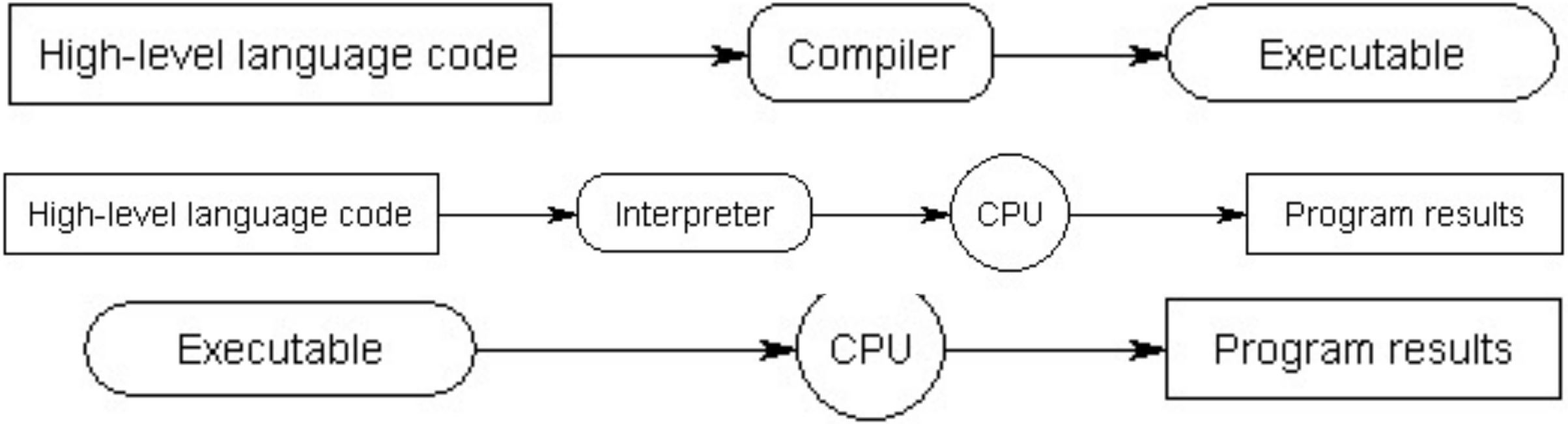
- return 0
  - 通知OS，本程序成功执行完毕。
  - 是main block的最后一行
- 注意
  - 每一个声明需要分号结束（预处理命令和{}除外（如果是定义class的时候，{}也要跟着分号））
  - 忘记分号，是新手常犯错误

The Compilation Process

Our language v.s. binary language the computer used

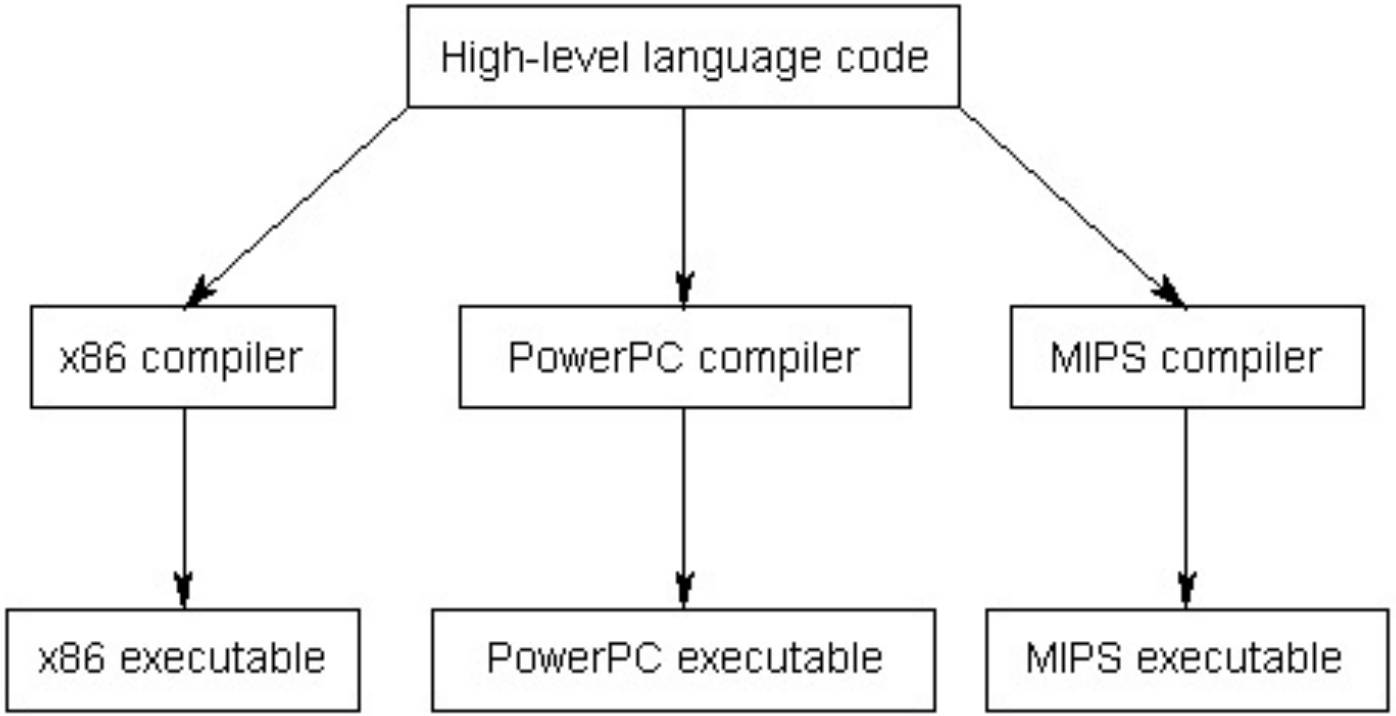
C++ is like natural language

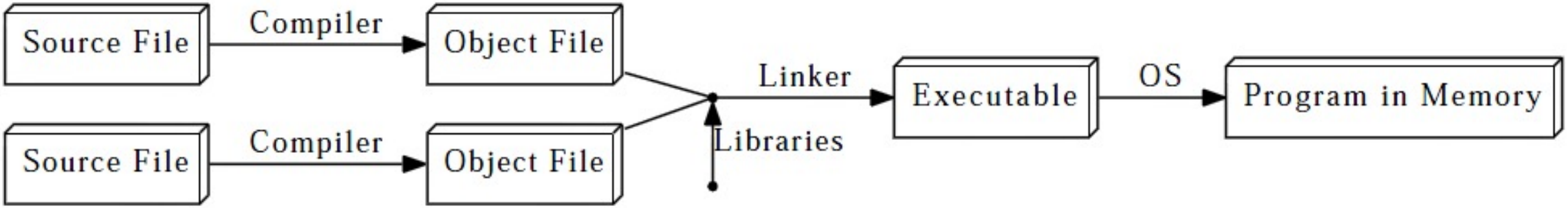
Compiler: make computer understand C++



The Compilation Process

Compiler: make computer understand C++



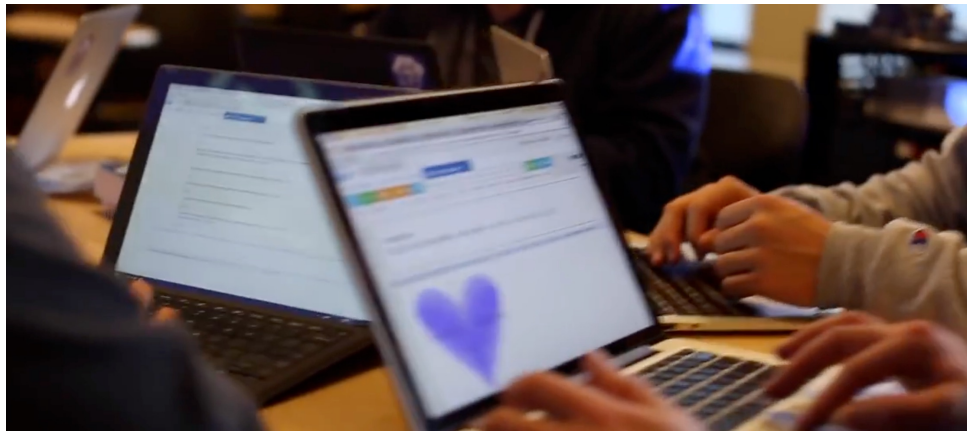


How to construct your virtual world?

- Every creative activity needs tools: a sheet of paper and a pencil?

- **Running** the code is the **only method** of finding out whether it's correct.
  - a computer equipped with some additional tools.
- A standard text editor and command-line compiler tools? May talk this later.

- An **IDE** is better.
- Or On-line tools?



IDE (Integrated Development Environment)

- A software: a code **editor, a compiler, a debugger**, and a graphical user interface (GUI) builder.
  - consume a lot of resources and, frankly speaking, you probably don't need most of the functions they can perform.

- If using on-line tools: an Internet browser + Internet access. But …

- Choose the one that's more convenient for you.

eclipse

Code::Blocks

Visual Studio

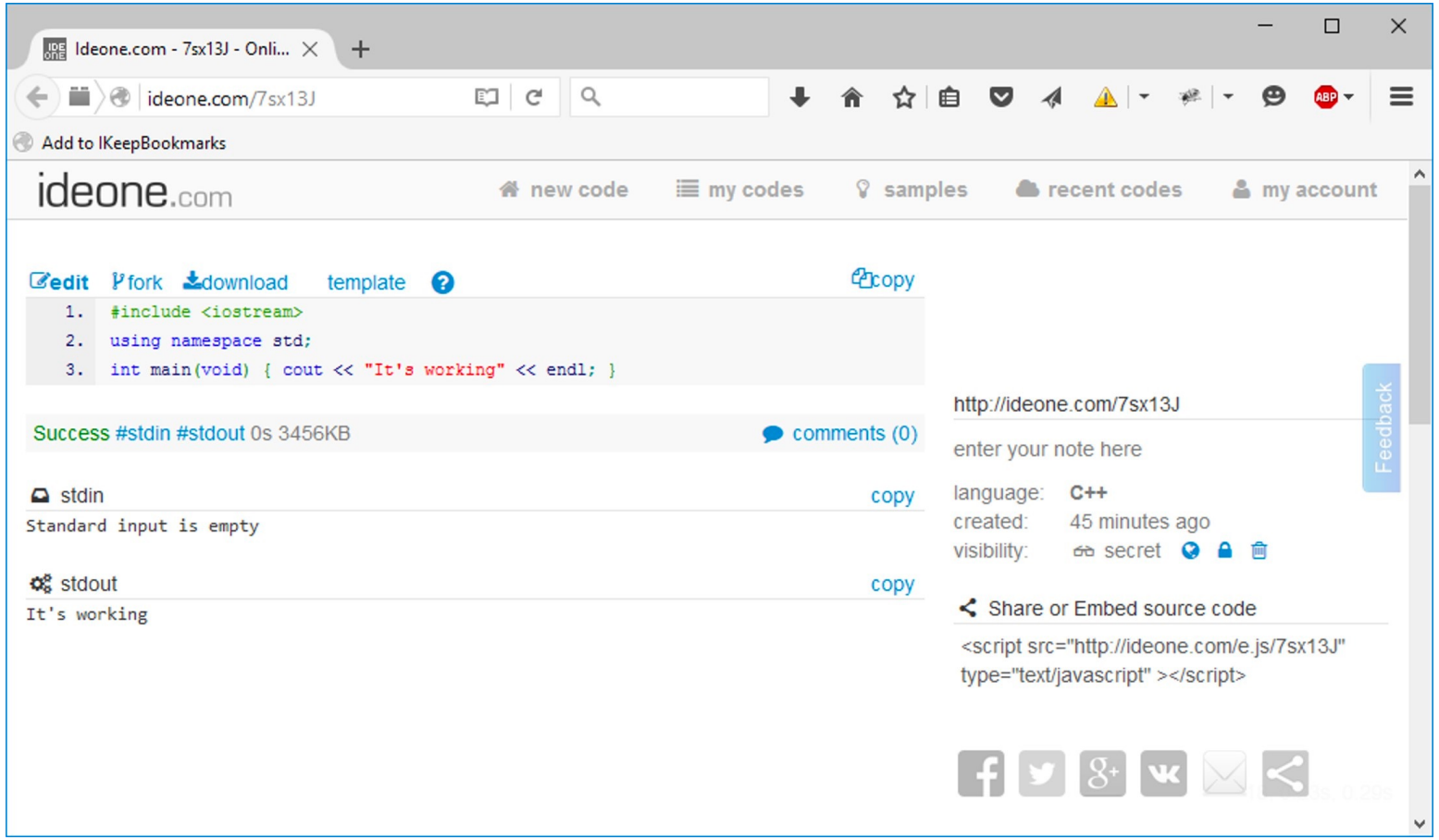
NetBeans

Visual Studio Code

[Installing an Integrated Development Environment (IDE)]

On-line tools: ideone

- http://ideone.com
- http://cpp.sh



编译你的第一个程序

- lab01_IDE_vscode_helloworld.pptx
- lab01_IDE_VC_Win32ConsoleApplication.pptx

- LearnCpp.com

C and C++’s philosophy能力与责任

- Underlying design philosophy: “**trust the programmer**”
  - Wonderful
    - compiler will not stand in your way if you try to do something unorthodox that makes sense,
  - Dangerous
    - compiler will not stand in your way if you try to do something that could produce unexpected results.
    - That is one of the primary reasons why knowing **what you shouldn’t do** in C/C++ is almost **as important as knowing what you should do** -- because there are quite a few pitfalls that new programmers are likely to fall into if caught unaware.

Reference Books

1. **C++ Primer, 5th version**
2. The C++ Programming Language. (more advance than 1)
3. The C++ Standard Library – A Tutorial and Reference
4. **Teach Yourself C++ in One Hour a Day**

5. **Code complete 2nd**
6. Clean Code A Handbook of Agile Software Craftsmanship

Reference Courses

- cpp for school http://www.cppforschool.com
  - simpler and with assignments, projects, quiz and papers.
- LearnCpp.com http://www.learncpp.com
  - more detail explanations than cpp for school

- [online course & IDE]: [cpp在线中文教程](https://www.runoob.com/cplusplus/cpp-tutorial.html)，包括主流操作系统下g++和Visual C++的设置. 提供在线编译运行，可见编译错误提示，程序输出等
- [online course & IDE](https://www.dotcpp.com),大量练习题，在线提交，可以获取3种状态：编译失败，运行结果错误，成功。

Useful Links

- http://www.cplusplus.com

- [代码打包工具]: Visual C++代码打包工具, 可以自行调整.
- [总结]: C++知识体系, 总结的很好，包括一些高级内容.

Academic Integrity

- Honest work is required of a scientist or engineer.
- Integrity is the key for everything!!!

- Discussion is permitted.
- **Everything you turn in must be your own work.**
- Cite your sources, explain any unconventional action.

- **If you have a question, ask.**