四川大學
SICHUAN UNIVERSITY

Object Oriented Programming—C++

Lecture 1: Introduction

Qijun Zhao

College of Computer Science

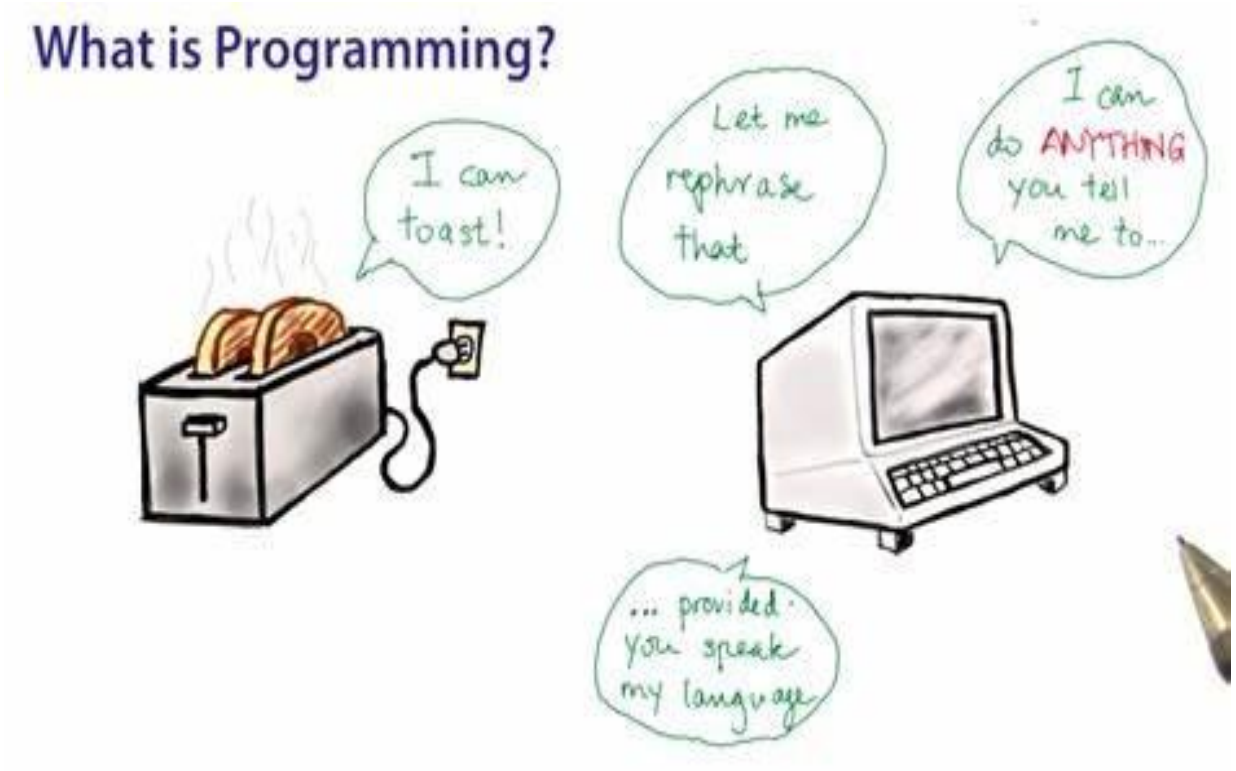
Sichuan University

Spring 2023

Something you should already know

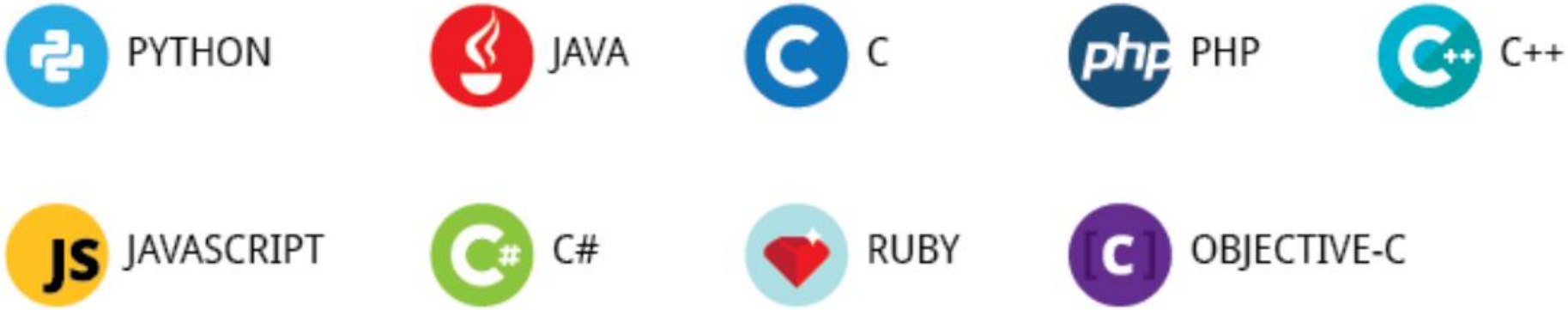
What does programming mean to ...

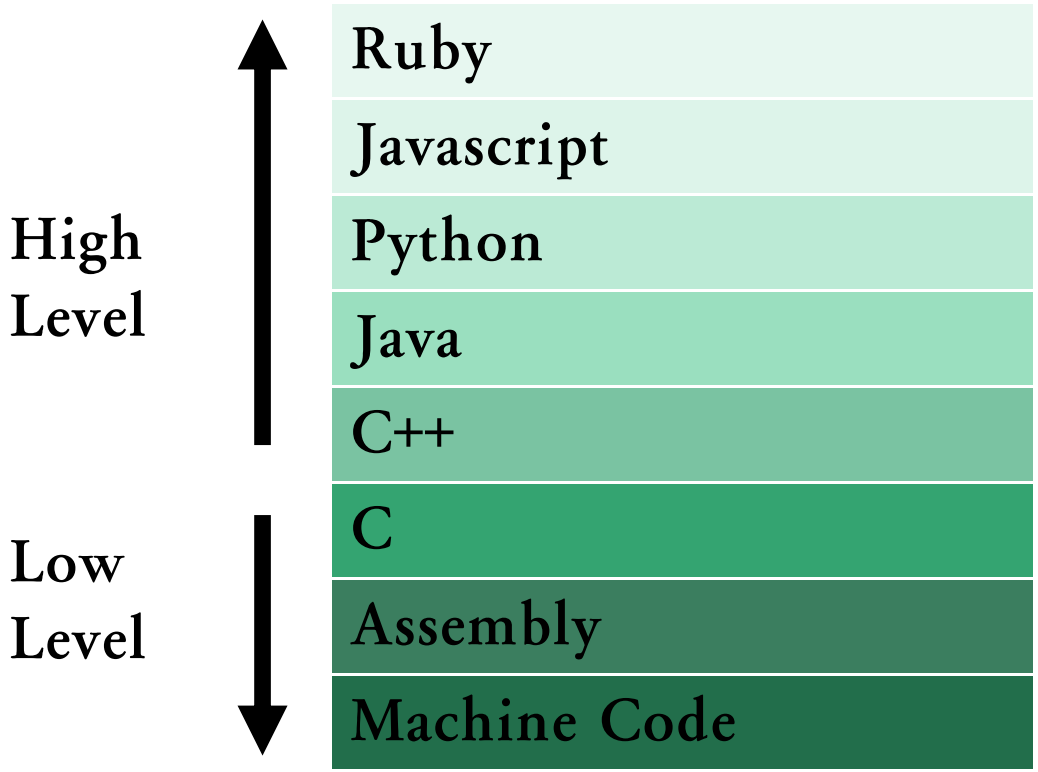
- What does Programming mean to the world?

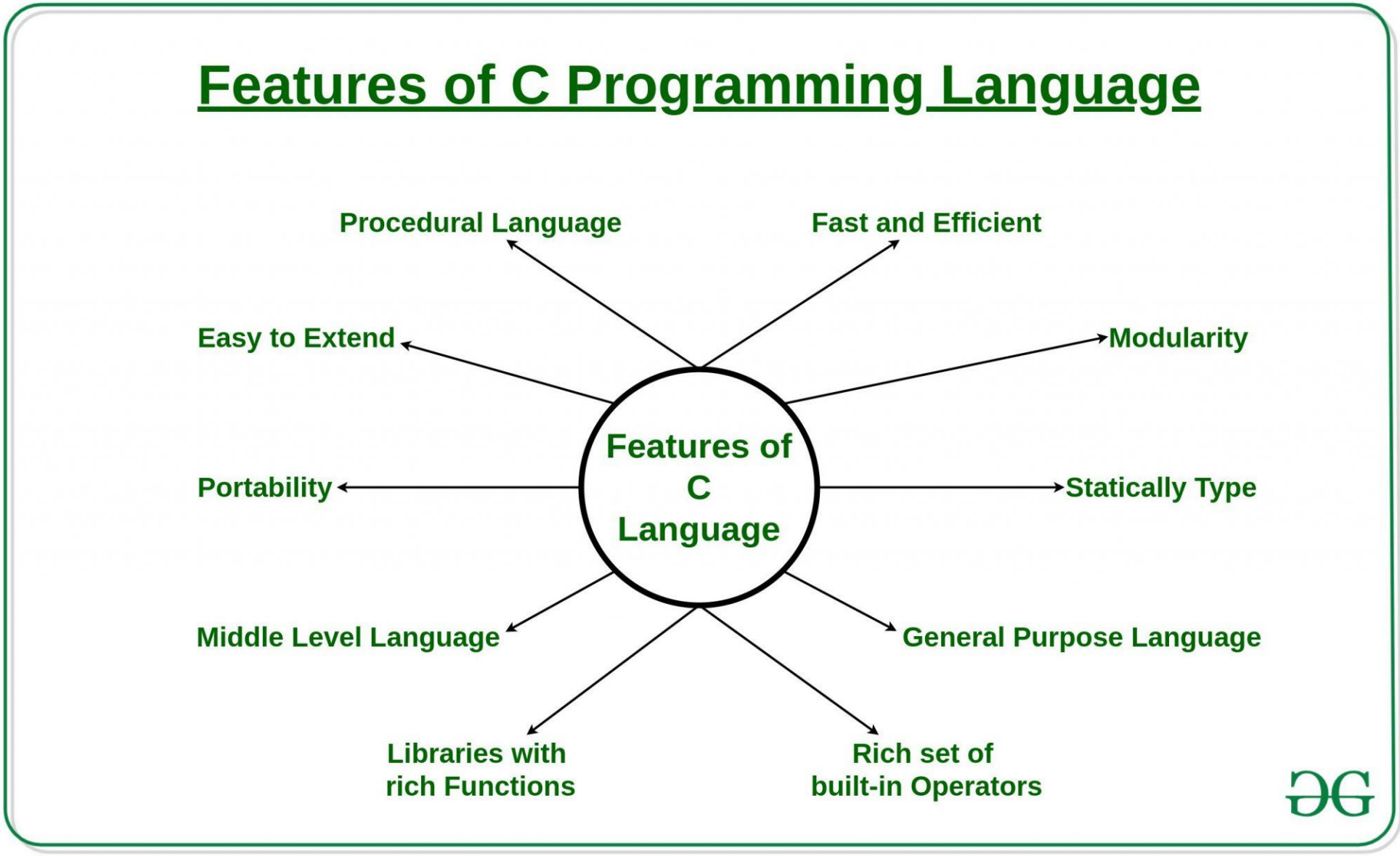


- What does Programming mean to you?

The programming languages in the world






Features of C Programming Language
Procedural Language
Fast and Efficient
Easy to Extend
Modularity
Features of C Language
Portability
Statically Type
Middle Level Language
General Purpose Language
Libraries with rich Functions
Rich set of built-in Operators

Interpreted Language vs. Compiled Language

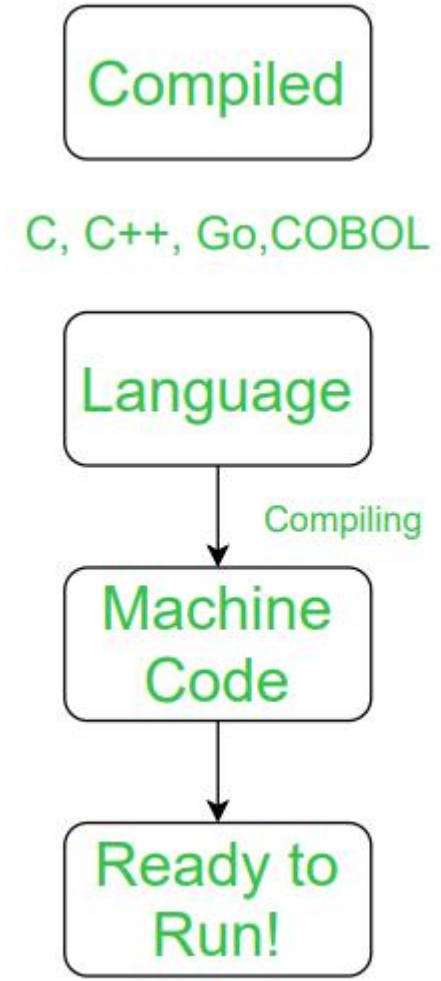
An interpreted language is a programming language that is generally interpreted, without compiling a program into machine instructions. It is one where the instructions are not directly executed by the target machine, but instead, read and executed by some other program.

Interpreted language ranges – JavaScript, Perl, Python, BASIC, etc.

Interpreted Language vs. Compiled Language

A compiled language is a programming language that is generally compiled and not interpreted. It is one where the program, once compiled, is expressed in the instructions of the target machine; this machine code is undecipherable by humans.

Types of compiled language – C, C++, C#, CLEO, COBOL, etc.



Difference between Compiled and Interpreted Language

NO.	COMPILED LANGUAGE	INTERPRETED LANGUAGE
1	A compiled language is a programming language whose implementations are typically compilers and not interpreters.	An interpreted language is a programming language whose implementations execute instructions directly and freely, without previously compiling a program into machine-language instructions.
2	In this language, once the program is compiled it is expressed in the instructions of the target machine.	While in this language, the instructions are not directly executed by the target machine.
3	There are at least two steps to get from source code to execution.	There is only one step to get from source code to execution.
4	In this language, compiled programs run faster than interpreted programs.	While in this language, interpreted programs can be modified while the program is running.
5	In this language, compilation errors prevent the code from compiling.	In this languages, all the debugging occurs at run-time.
6	The code of compiled language can be executed directly by the computer’s CPU.	A program written in an interpreted language is not compiled, it is interpreted.
7	This language delivers better performance.	This language example delivers relatively slower performance.
8	Example of compiled language C, C++, C#, CLEO, COBOL, etc.	Example of Interpreted language JavaScript, Perl, Python, BASIC, etc.

The 4 Stages of C Compilation

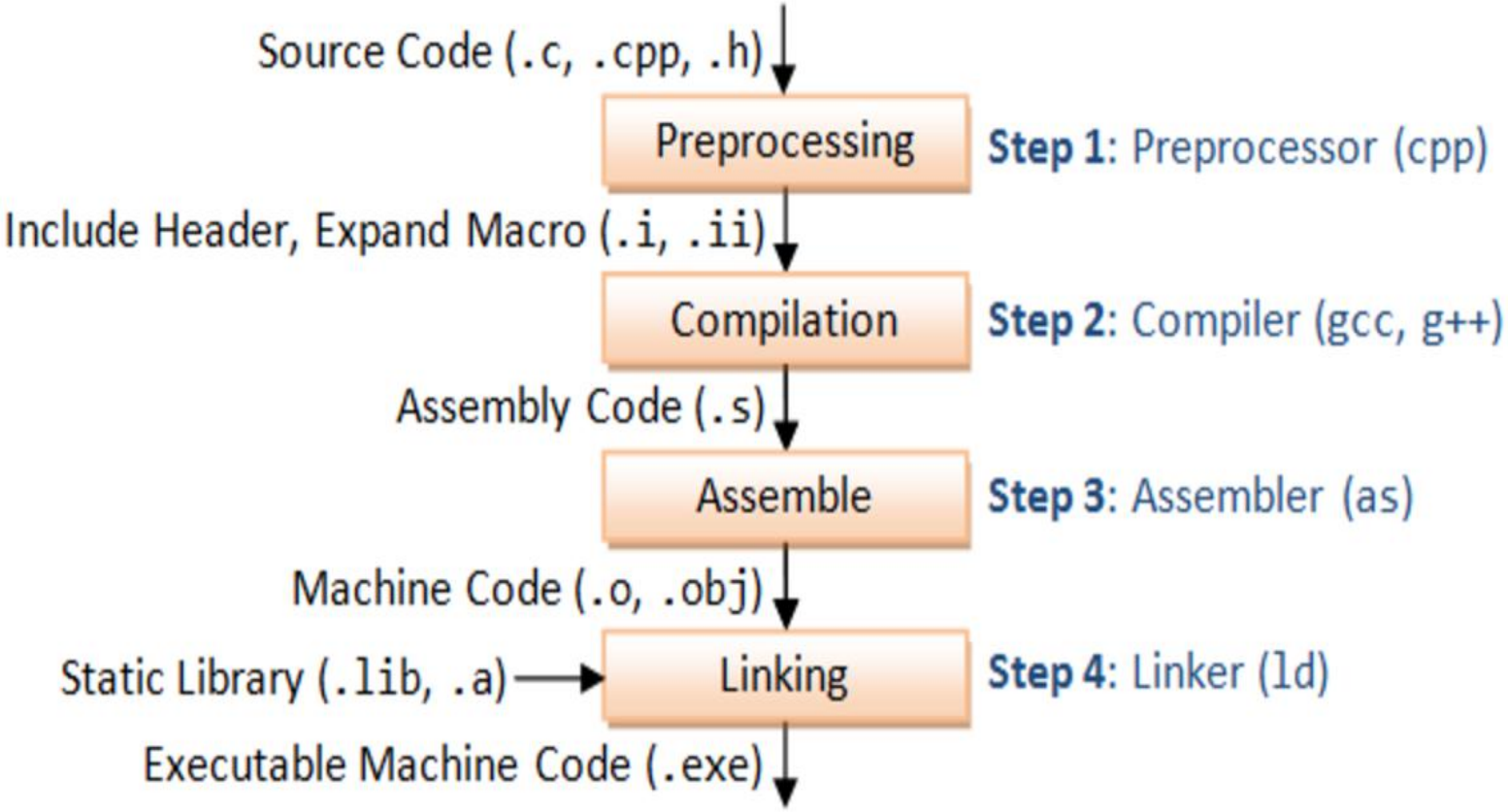
The compilation can split into 4 stages:

-Preprocessing

-Compilation

-Assembly

-Linking



About this course

Purpose of this course

- Exposure to standard c++ syntax and norms

- Learn key features of object oriented programming (OOP)

- Gain familiarity with powerful features of the stl

- Practice using industry standard coding tools such as ssh

Teaching team

- Lecturer
  - Prof. / Dr. Qijun Zhao

- Teaching assistants
  - Hongmin Shao, A master student
  - Zhonghui Zhang, A junior undergraduate student (in top-notched students program)
  - Yuchuan Deng, A sophomore undergraduate student (in top-notched students program)

Course Overview

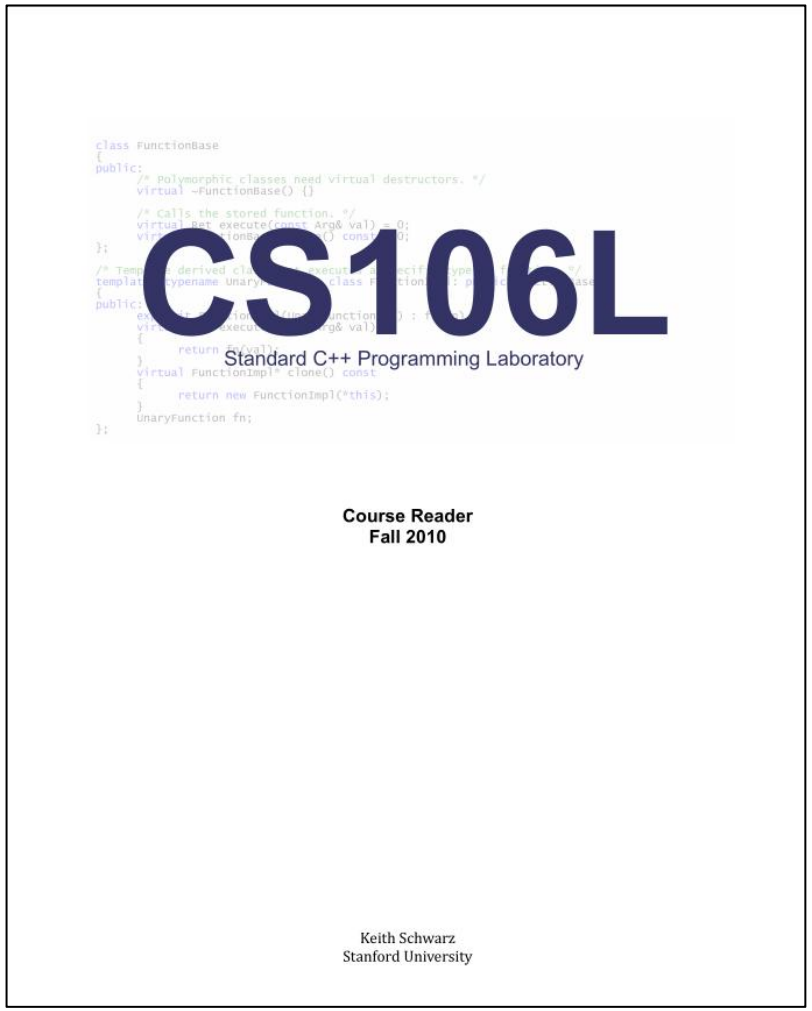
周次	日期	理论课主题	上机课（4-13周）	内容
1（校历第2周）	2月28日	Introduction		
2（校历第3周）	3月7日	Types and Structs Initialization and References Streams Containers Iterators and Pointers	To be announced at Week 4	Features of C++
3（校历第4周）	3月14日			
4（校历第5周）	3月21日			
5（校历第6周）	3月28日			
6（校历第7周）	4月4日	Quiz I & Mid-Term Review		
7（校历第8周）	4月11日	Classes Template Classes Template Functions Functions and Algorithms Operator Overloading Special Member Functions		Object Oriented Programming (OOP)
8（校历第9周）	4月18日			
9（校历第10周）	4月25日			
10（校历第11周）	5月2日			
11（校历第12周）	5月9日			
12（校历第13周）	5月16日	Quiz II		
13（校历第14周）	5月23日	Move Semantics		Advanced Features of C++
14（校历第15周）	5月30日	Type Safety		
15（校历第16周）	6月6日	Final Review		

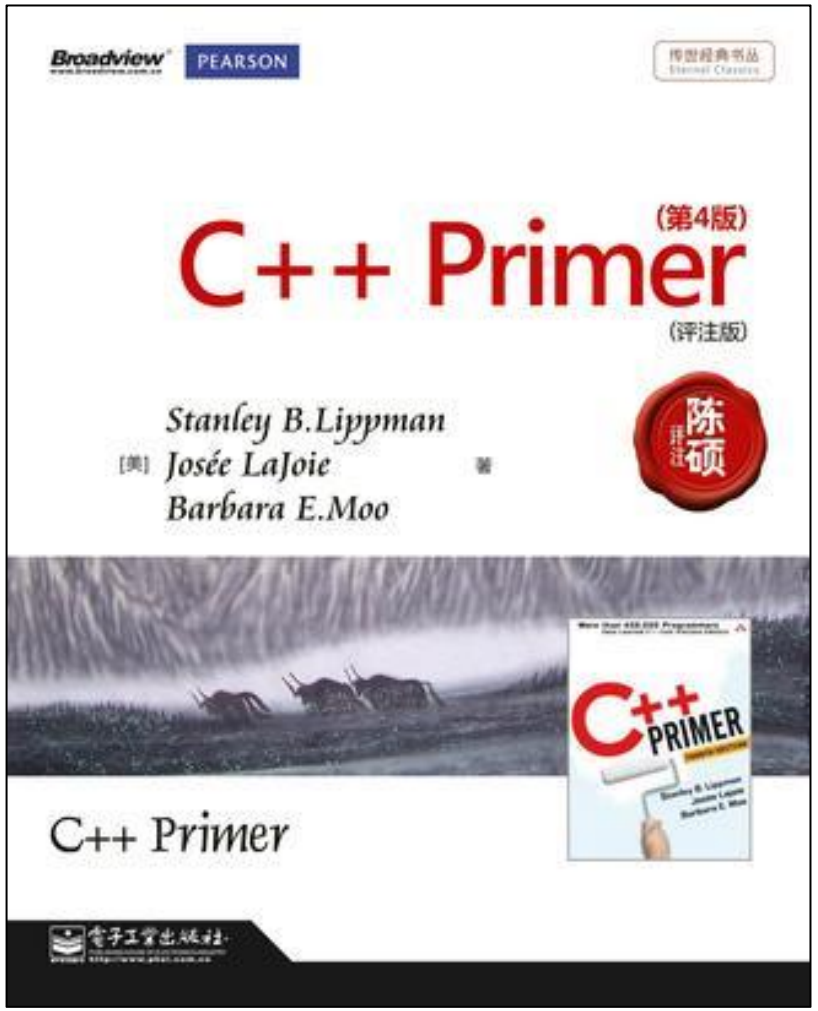
Acknowledgement: Materials of this course are mainly adapted from CS106L @ Stanford.

Course Grade Composition

Attendance & On-class Performance	10%
Assignment	20%
Quiz I	20%
Quiz II	20%
Final Exam	30%

Reader





https://stackoverflow.com/questions/388242/the-definitive-c-book-guide-and-list

Why C++?

Why C++?

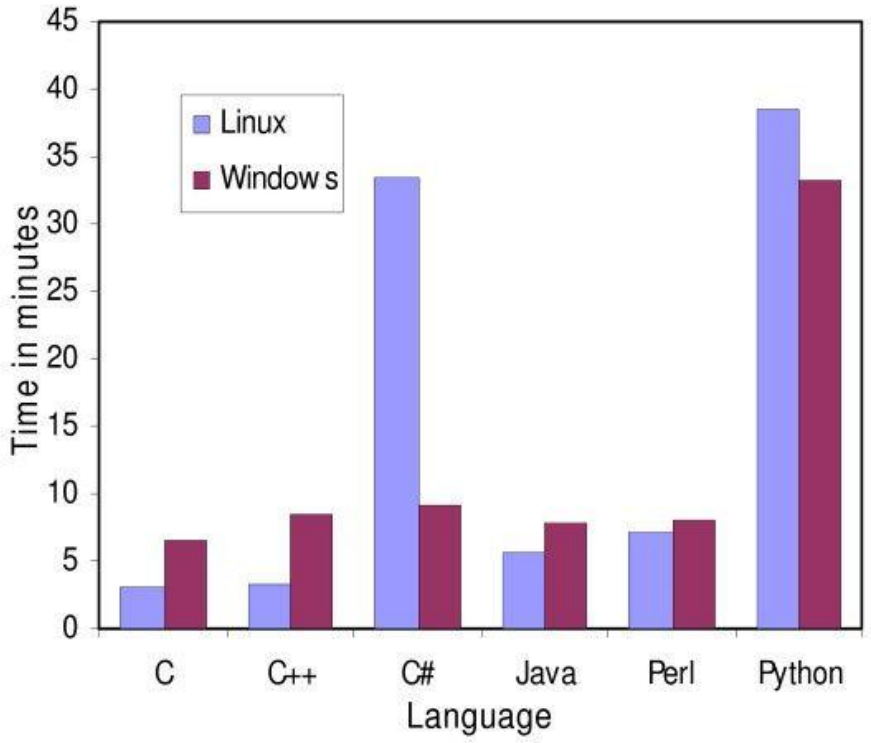
C++ is still a very popular language!

May 2021	Programming Language	Ratings	Chart Ratings
1	C	13.38%	
2	Python	11.87%	
3	Java	11.74%	
4	C++	7.81%	
5	C#	4.41%	
6	Visual Basic	4.02%	

Tiobe Index, 2021

Why C++?

FAST



High Level

Low Level

- Ruby
- Javascript
- Python
- Java
- C++
- C
- Assembly
- Machine Code

What is C++?

```cpp
#include <iostream>

int main() {
    std::cout << "Hello, world!" << std::endl;
    return 0;
}
```

What is C++?

```c
#include "stdio.h"
#include "stdlib.h"

int main(int argc, char *argv) {
    printf("%s", "Hello, world!\n");
    // ^a C function!
    return EXIT_SUCCESS;
}
```

What is C++?

```
#include "stdio.h"
#include "stdlib.h"

int main(int argc, char *argv) {
   asm(  "sub    $0x20,%rsp\n\t"                    // assembly code!
         "movabs $0x77202c6f6c6c6548,%rax\n\t"
         "mov    %rax,(%rsp)\n\t"
         "movl   $0x646c726f, 0x8(%rsp)\n\t"
         "movw   $0x21, 0xc(%rsp)\n\t"
         "movb   $0x0,0xd(%rsp)\n\t"
         "leaq    (%rsp),%rax\n\t"
         "mov    %rax,%rdi\n\t"
         "call   __Z6myputsPc\n\t"
         "add    $0x20, %rsp\n\t"
    );
    return EXIT_SUCCESS;
}
```

C++ History: Assembly

```
section     .text
global      _start                      ;must be declared  for linker (ld)

_start:                                 ;tell linker entry point

    mov     edx,len                     ;message length
    mov     ecx,msg                     ;message to write
    mov     ebx, 1                      ;file descriptor (stdout)
    mov     eax, 4                      ;system call number (sys_write)
    int     0x80                        ;call kernel
    mov     eax, 1                      ;system call number (sys_exit)
    int     0x80                        ;call kernel

section     .data
msg     db   'Hello, world!',0xa        ;our dear string
len     equ $ - msg                     ;length of our dear string
```

C++ History: Assembly

- Unbelievably simple instructions
- Extremely fast (when well-written)
- Complete control over your program

Why don’t we always use Assembly?

Assembly looks like this

```
section     .text
global      _start                          ;must be declared  for linker (ld)

_start:                                     ;tell linker entry point

    mov     edx,len                         ;message length
    mov     ecx,msg                         ;message to write
    mov     ebx,1                           ;file descriptor (stdout)
    mov     eax,4                           ;system call number (sys_write)
    int     0x80                            ;call kernel
    mov     eax,1                           ;system call number (sys_exit)
    int     0x80                            ;call kernel

section     .data
msg     db  'Hello, world!',0xa             ;our dear string
len     equ $ - msg                         ;length of our dear string
```

C++ History: Assembly

Drawbacks:

- A LOT of code to do simple tasks
- Very hard to understand
- Extremely unportable (hard to make work across all systems)

Invention of C

Problem: computers can only understand assembly!

- **Idea:**

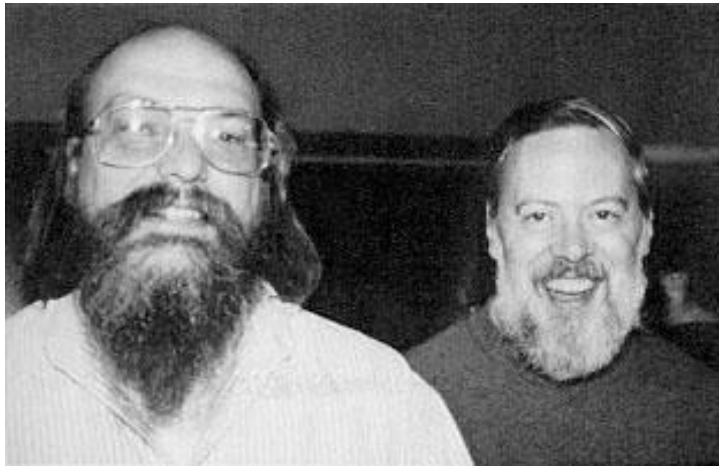
- Source code can be written in a more intuitive language

- An additional program can convert it into assembly

- This additional program is called a **compiler**!

C++ History: Invention of C

- T&R created C in 1972, to much praise
- C made it easy to write code that was
  - Fast
  - Simple
  - Cross-platform



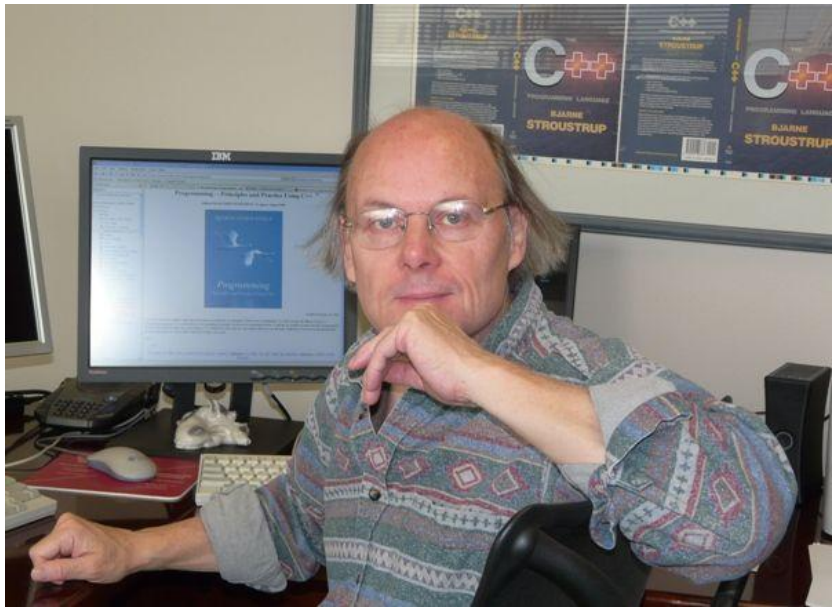
Ken Thompson and Dennis Ritchie, creators of the C language.

C++ History: Invention of C

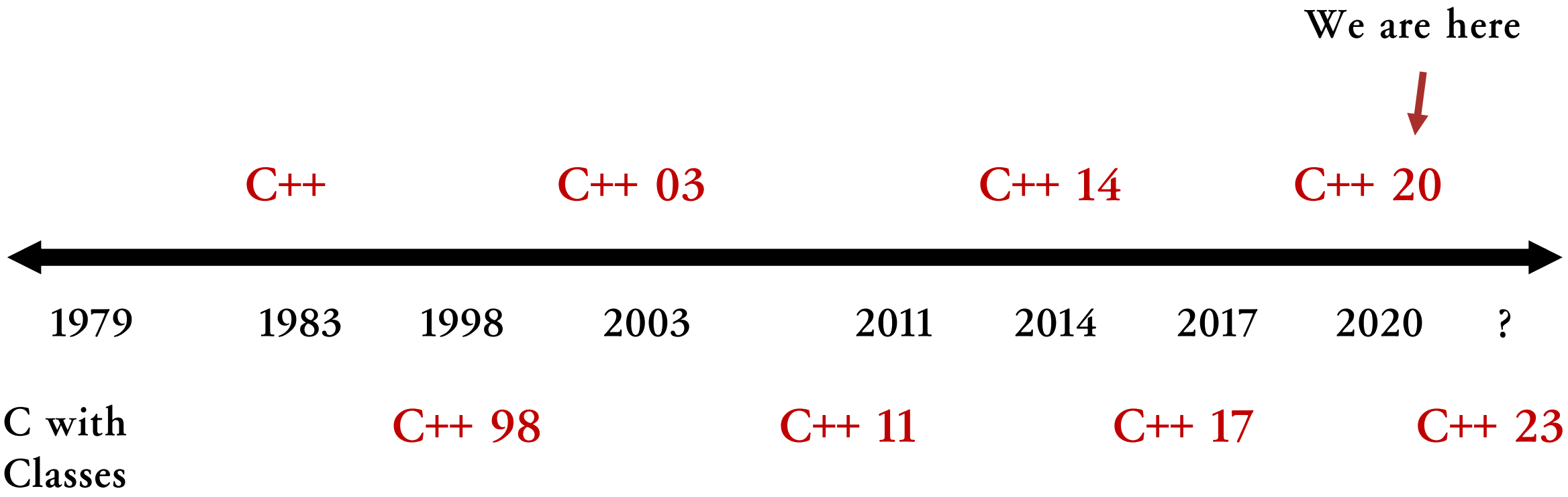
- C was popular because it was simple.
- This was also its weakness:
  - No objects or classes
  - Difficult to write generic code
  - Tedious when writing large programs

C++ History: Welcome to C++!

- In 1983, the beginnings of C++ were created by Bjarne Stroustrup.
- He wanted a language that was:
  - Fast
  - Simple to use
  - Cross-platform
  - Had high-level features



C++ History: Evolution of C++



Design Philosophy of C++

- 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

Design Philosophy of C++

- 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

Design Philosophy of C++

- 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

But... What is C++?

C++: Basic Syntax & the STL

Basic syntax

- Semicolons at EOL
- Primitive types (ints,doubles etc)
- Basic grammar rules

The STL

- Tons of general functionality
- Built in classes like maps, sets, vectors
- Accessed through the namespace std::
- Extremely powerful and well-maintained

How to learn and practice?

四川大學
SICHUAN UNIVERSITY

Coding for love,
Coding for the world

Qijun Zhao

qjzhao@scu.edu.cn