Welcome to CS106L

Original from : Mike Precup (mprecup@cs.stanford.edu)
ENJMIN Edition 2016

Lecture 0: Syllabus/C++ Overview

- Administrivia
- What is CS106L?
- The history and philosophy of C++
- Why learn C++?

Lecture 0: Syllabus/C++ Overview

- The history and philosophy of C++
- Why learn C++?

History: Assembly

- Extremely simple instructions
- Requires lots of code to do simple tasks
- Can express anything your computer can do
- Hard to read, write

History: Assembly

Here's an example of a "hello world" program in 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
```

History: C

- C was created in 1972 to much praise
- C made it easier to write fast cross platform code

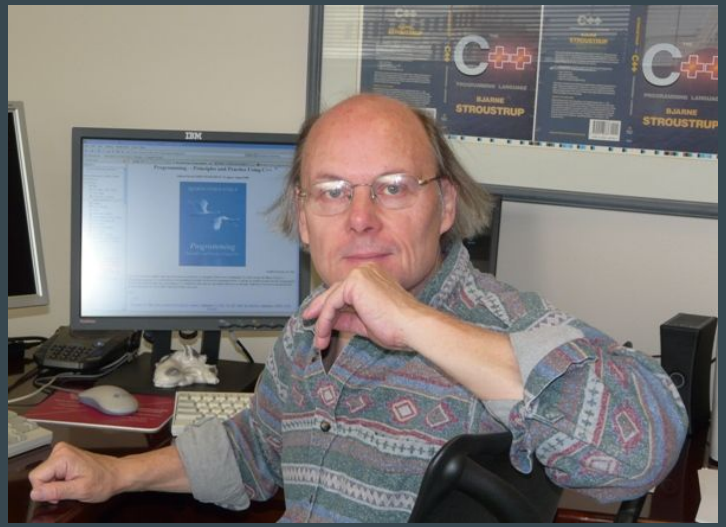
Ken Thompson and Dennis Ritchie, creators of the C language

History: C

- C was popular since it was simple
  - Not much useless syntax
  - Runs extremely fast
  - Runs on any computer with a C compiler
- C was eventually criticized since it was simple
  - No objects or classes (e.g. no maps)
  - Sometimes even simple tasks are extremely difficult to program

History

- C++ was created in 1983 by Bjarne Stroustrup, this guy:



Why make C++?

- Bjarne wanted a mix of:
  - High level language features
  - Speed
  - Portability
  - Usability
- C already had all of those except the features

C++ Through the Ages

- C with classes
- C++
- C++98
- C++03
- C++11
- C++14
- Soon, C++17!

Hello World in C++

```
#include <stdio.h>

#include <stdlib.h>

int main(int argc, char *argv[]) {

   puts("Hello world");

   return EXIT_SUCCESS;

}
```

Hello World in C++

```
#include <iostream>

using namespace std;

int main() {

   cout << "Hello World!" << endl;

}
```

Philosophy

- 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

Lecture 0: Syllabus/C++ Overview

- The history and philosophy of C++
- Why learn C++?

Why Learn C++?

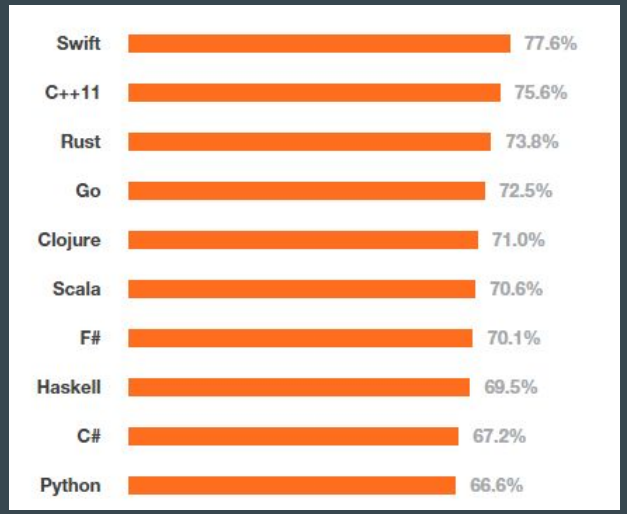
It's popular:

Mar 2016	Mar 2015	Change	Programming Language	Ratings	Change
1	2	^	Java	20.528%	+4.95%
2	1	v	C	14.600%	-2.04%
3	4	^	C++	6.721%	+0.09%
4	5	^	C#	4.271%	-0.65%
5	8	^	Python	4.257%	+1.64%
6	6		PHP	2.768%	-1.23%
7	9	^	Visual Basic .NET	2.561%	+0.24%
8	7	v	JavaScript	2.333%	-1.30%

(TIOBE index, March 2016)

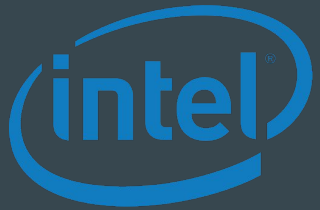
Why Learn C++?

It's popular:



(Most loved languages, StackOverflow 2015)

Why C++: Users (companies)

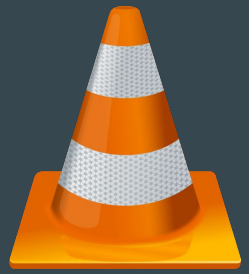


Why C++: Users (web browsers)





Why C++: Users (software)

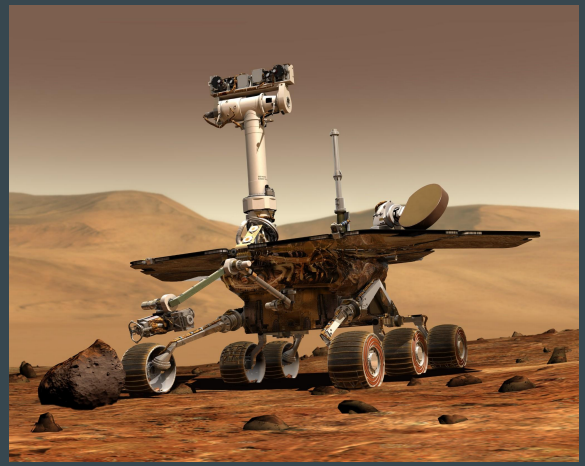


Why C++: Users (games)

Why C++: Users (cool things)

The F-35 Lightning II (Joint Strike Fighter) relies extensively on C++

The Spirit rover was operational for over 6 years when the mission was only planned to run for around 3 months



“One of the things I really like about programming languages is that it's the perfect excuse to stick your nose into any field. So if you're interested in high energy physics and the structure of the universe, being a programmer is one of the best ways to get in there. It's probably easier than becoming a theoretical physicist”

-Bjarne Stroustrup

The Big Picture

- This class is partially about features, and partially about style
- Using C++'s features well will clean up your code significantly
- Features were added to make your life easier

The Big Picture

```
template <typename T>

Widget<T>& Widget<T>::operator=(const Widget& other) {

    Widget copy(other);

    swap(copy);

    return *this;

}
```