Programming Abstractions

CS106B

Cynthia Lee
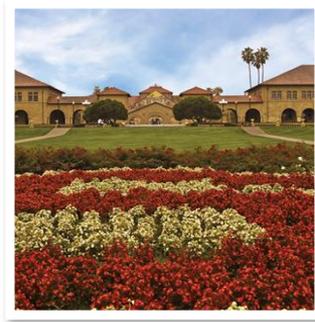
Today's Topics

Introducing C++

- Hello, world!
  - › Code with me at codestepbystep.com
- firstprogram.cpp
  - › In QT Creator (the IDE for our class)
  - › Function prototypes
  - › <iostream> and cout
- Hamilton example
  - › C++ strings and streams
  - › <iostream> and cin/cout
  - › "simpio.h" and getLine()

C++ Basics: Hello, world!

GO TO
CODESTEPBYSTEP.COM

C++ Basics: Making functions

DEMONSTRATION IN QT CREATOR

A simple C++ program (Error)

firstprogram.cpp

```cpp
#include <iostream>
#include "console.h"
using namespace std;

int main(){
    cout << "|-5| = " << absoluteValue(-5) << endl;
    return 0;
}

int absoluteValue(int n) {
    if (n < 0){
        return -n;
    }
    return n;
}
```

A simple C++ program (Fixed v.1)

firstprogram.cpp

```cpp
#include <iostream>
#include "console.h"
using namespace std;

int absoluteValue(int n) {
    if (n < 0){
        return -n;
    }
    return n;
}

int main(){
    cout << "|-5| = " << absoluteValue(-5) << endl;
    return 0;
}
```

A simple C++ program (Fixed v.2)

firstprogram.cpp

```cpp
#include <iostream>
#include "console.h"
using namespace std;

int absoluteValue(int n);

int main(){
    cout << "|-5| = " << absoluteValue(-5) << endl;
    return 0;
}

int absoluteValue(int n) {
    if (n < 0){
        return -n;
    }
    return n;
}
```

Design Question: Why does C++ have the function prototype syntax?

In other words, why not just have a rule that you must set up the ordering so you define your functions before using them, as in the "FIXED 1" example?

A. C++ could have done that, but such a rule would be *too cumbersome* for programmers to follow.

B. C++ could have done that, but good programming *style* dictates "top-down" approach that logically puts main() first and helper functions it calls to follow.

C. C++ could *not* have done that, because sometimes there is *no way* to order the functions so that all functions are defined before being used.

D. Other/none/more than one of the above

[Answer] Why does C++ have the function prototype syntax?

(A) and (B) The rationales behind choices (A) and (B) (previous slide) are correct

› May or may not have been enough to compel the language designers to introduce the function prototype feature

(C) is true—there are cases where you simply cannot rearrange the ordering of functions to avoid all cases of use before definition

› e.g., mutual recursion

Which came first, the chicken or the egg?

(this code is just for fun, for now—we'll cover recursion in depth in a few weeks!)

```cpp
#include<iostream>
#include "console.h"
using namespace std;

void go(int n);
void stanford(int n);

int main(){
    int repetitions = 5;
    go(repetitions);
    return 0;
}
```
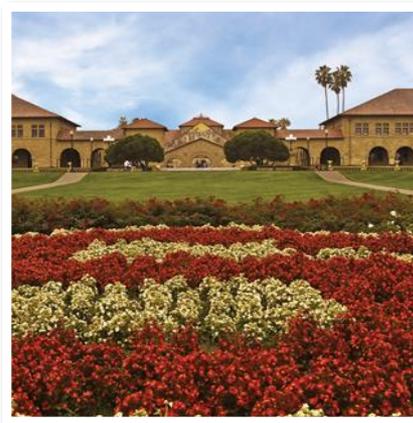
```cpp
void go(int repetitions) {
    if (repetitions <= 0) return;
    cout << "Go!" << endl;
    stanford(repetitions - 1);
}


void stanford(int repetitions) {
    cout << "Stanford!" << endl;
    go(repetitions);
}
```

Streams in C++

Hamilton Example
iostream (C++ Standard)
simpio (Stanford)

Hamilton Code Demo:
What essential skills did we just see?

- You can read and write input/output with:
  - › `cout, cin` (`<iostream>`)
  - › `getInteger()`, `getLine()`, etc print a message before waiting for input (`"simpio.h"`)
- cin and cout use the >> and << operators, respectively
  - › Remember: the arrows point in the way the data is "flowing"
  - › These aren't like HTML tags or C++ parentheses () or curly braces {} in that they don't need to "match"
- Good style: "`static const int`" to make int constants
  - › No "magic numbers"!
  - › Works for other types too ("`static const double`")