CS 142
C++ Basics

Rhodes College

Announcements

- Program 4 due Tuesday, 3/17th by 11:55pm

A bit of history

- Transitioning to C++ programming language
- C++ history
  - Offshoot of the language C (1969-73)
  - Developed by Bjarne Stroustroup starting in 1979.

Python vs. C++

Python

- Interpreted
- Dynamically typed
  - def square(x)
    return x * x
  - You can call square("two")

C++

- Compiled
- Statically typed
  - Program won't even run if you tried the square example – C++ compiler will flag it as an error

Different syntax; same control structures.
Differences are cosmetic.

Both are object-oriented languages.
C++ is similar to Python in that you can do non-object oriented stuff as well.
Going to introduce C++ for a few weeks without any OO, then put in the OO later.

Writing your first C++ program

- Use handout

Parts of the code

```
#include <iostream>
using namespace std;

int main()
{
      cout << "Hello World\n";

      return 0;

}
```

Like an import statement
Unlike Python, you don't get print and input and all for free
Must have this line in every program.

Parts of the code

```
#include <iostream>
using namespace std;

int main()
{
      cout << "Hello World\n";

      return 0;

}
```

Kind of like
from X import * in Python
Also needs to be in all your programs.

Parts of the code

```
#include <iostream>

using namespace std;

int main()
{
      cout << "Hello World\n";

      return 0;

}
```

Instead of def main():

Instead of print("Hello World")

Technically, in C++ the main function has to return a value because it is declared as "int main" which means "main function should return integer data type"
0 tells the operating system that the program ran successfully.

Parts of the code

```
#include <iostream>

using namespace std;

int main()
{
      cout << "Hello World\n";

      return 0;

}
```

C++ uses semicolons at the end of each statement.

Parts of the code

```
#include <iostream>

using namespace std;

int main()
{
      cout << "Hello World\n";

      return 0;

}
```

Cout does not put a newline at the end. Use \n or endl.

Parts of the code

```
#include <iostream>

using namespace std;

int main()
{
      cout << "Hello World\n";

      return 0;

}
```

Indentation (technically) does not matter. Instead, what matters is the curly braces. You should still indent with curly braces because it makes your program easier to read!

Data Types

- You must declare your variables before you use them!!!
  - int
  - float
  - string [to use strings #include<string>]
  - We'll talk about other types later.
  - All variables are local, like Python.
- Can declare them ANYTIME before using them, but when in doubt, declare them at the top of the function.

Variables

- Declaring variables

```
int x, y;  //declares 2 integer types with names x and y
```

- Assigning values to variables

```
x = 5;
y = 7;
```

- Combining into single statement

```
int x = 5;
int y = 7;
```


Getting Input

- Use cin keyword

```
cin >> x >> y;
```

- You need to write your own prompt with cout.
- Cin can input multiple things at the same time.
- With strings, only reads one word at a time, not a whole line.
- Like Python, your program MIGHT crash/MIGHT keep running with bad data if you try to cin an int, but type a float.

C++ Demo Program

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    int x, y;
    string s;
    cout << "Type in two numbers: ";
    cin >> x >> y;
    cout << x + y << endl;


    cout << "Enter a string: ";
    cin >> s;
    cout << s << endl;

    return 0;
}
```