Basics of C++ and the Unix command prompt

Joel Willoughby

COP 3503

Outline

Lab Setup

- Start off with a discussion of important concepts
- Then, an in-lab assignment
- You can use resources (your book, discuss with others, the current lab slides etc.) to finish the lab
- When you are finished, let me know. I will check you have met the requirements, then you can leave (Or stay and work on whatever you need to)

Outline

What is C++

- It is a programming language
- Very similar syntax to Java (both were based on C)
- Has support for both OOP and traditional C-style
- The language you will be using in this course

Hello World in C++

```
1 #include <iostream>
2
3 int main(int argc, char ** argv)
4 {
5     std::cout << "Hello World!";
6     std::cout << std::endl;
7     return 0;
8 }
```

- Running this prints Hello World! to the console
- Let's break it down

The cout command

- Standard syntax is:

```
1 std::cout << thing_to_be_printed << std::endl;
```

- Can use variables, strings, etc. i.e.

```
1 int i=25;
2 std::cout << "i = " << i << std::endl;
```

will print i = 25

- cout returns another instance of cout, this is why chaining is possible

- Similar to the System.out.println(""); method in Java

The cin command

- Standard syntax is:

```
1 int i;
2 std::cin >> i;
```

the variable i will then have the value the user inputs

- Note the similarities to cout
- Similar to Java's Scanner class

Why all the std?

- When Dr. Nemo showed you “Hello World”, there was only cout, no std.
- std is a namespace. Think a box (namespace) on a table (global). The namespace tells the compiler where to look for a name.
- Dr. Nemo had:

```
1 #include <iostream>
2 using namespace std;
3
4 int main(int argc, char ** argv)
5 {
6     cout << "Hello World!";
7     cout << endl;
8     return 0;
9 }
```

- Using namespace tells the compiler where to look for everything. Think dumping the box on the table.

#include directive

```
1 #include <iostream>
2 #include "myheader.hpp"
```

- Included at the top of your files.
- Allows you to include “header” files, both user and system defined
  - Quotes, “”, for locally defined headers.
  - Brackets, <>, for C++ libraries (typically).
- Without it, the compiler doesn’t know what your code is using
- Similar to Java import

int main()

```
1 int main(int argc, char ** argv)
2 {
3     return 0;
4 }
5 int main(int argc, char* argv[]){return 0;}
6 int main(){return 0;}
```

- Every executable program needs a main (and only one main)
- This is where you program begins execution
- argc and argv are used for command line arguments
- Why return 0? A value of 0 indicates to the operating system the program executed normally

Logic

- Loops, ifs, and function syntax are almost identical to Java
- One difference is C++ allows integer values in place of booleans for conditionals. 0 is false, everything else is true.
- The following, then, is an infinite loop:

```
1 while(1)
2 {
3     //Do stuff
4 }
```

- Be careful with this feature

Outline

Unix Environment

- There are many ways of compiling
- Today, we will compile using the unix command line
  - Note that this is how I will grade projects, so make sure they compile from the command line in an ubuntu environment
- We will use the g++ compiler. It comes with any linux distro (also Mac OS)

Demo Time

This is easiest shown through a demo

Outline

Any questions

????