C++ Week 2

Karl Statz

Variables

```
int x = 10;
long z = 1000000;
double y = 1.1111111;
float f = 1.11;
```

functions

```
// function that returns nothing and takes in no arguments
void foo();

// function that returns something
int foo();

// overloaded function
int foo(int x);

// overloaded foo with a default argument
int foo(int x, int y = 10);
```

collections

```
//arrays
//fixed sized collection
int x[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

//vector (make sure you include <vector>)
//a variable sized, indexed collection
std::vector<int> myInts = std::vector<int>();
```

Local Development

CMake

Cmake is a build system for building c/c++ projects. It's primary job is to generate the appropriate make file for a certain project. I chose it because, unlike vanilla make, it has easy to remember syntax and is generally human readable.

Best Practices

1. create a build/ directory adjacent to your src/ directory. This keeps any build cruft contained and easy to delete
2. run all cmake commands from the build/ directory. running cmake ../ runs the command in the parent directory
3. add the build/ directory to your .gitignore so build artefacts arent committed.
4. after you run cmake and generate the build scripts, you compile your code with running ‘make‘ in the build directory