Functions
Feb. 10, Ch 4-5
Functions

So far we have been writing code inside main() without understanding some parts of it.

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

int main() {
    cout << "Hello world!" << endl;
    return 0;
}
```

copy paste this, else computer throws fit.
Dunno what this does but I can forget it and computer doesn't care.

Why zero?
Functions

Can think of methods as packaging multiple commands into one
Functions

We have used functions before, such as sqrt(), pow() or possibly round()

You can also create your own similar to creating variables by:

(1) declaring the function
(2) defining what the function does

(See: sayHi.cpp)
Functions

```c
int sayHi(); // Function declaration (put before main or any other definition)

int main()
{
    sayHi();
    return 0;
}

int sayHi()
{
    cout << "Howdy, I'm a computer!\n";
    return 0;
}
```
Functions

Functions, like variables, have types (int, double, char, etc.)

We call them the return value, as it is what the function will become after being finished.

For example: sqrt(4) will become 2.0 (double) when it is finished.

(See: addition.cpp)
Functions

The return statement value must be the same as the return type (or convertible) (See addition2.cpp)
Functions

You can actually have multiple functions with the same name, as long as the arguments are different either by:
- a different amount of arguments
- different types of arguments

This is called **overloading** a function

(See overloading.cpp)
Functions

You can make functions return type `void`, but not variables (an empty variable? ehh...)

This means nothing is returned, so you will get an error if you say:

```c
void x();
... then ...
int y = x(); // x not an int! or anything!
```

A good use of void functions is to print out
Functions

(See maze.cpp)
Functions

It is important to note that the code will resume after the function call where it was used.

For example, sqrt(4) will return the value 2.0 where it was used and the rest of your code will continue.

Where does the maze code return to?
Functions

Multiple function uses/calls create a “stack” much like pancakes: every time you use a function, it will add another pancake.

When you return, the top pancake is removed.

main() is the bottom pancake.
Functions

How to make the person run?

RUN FOREST, RUN!!!

(See: runForest2.cpp)