Functions

A Function is a section of code that can be used or called by a different section of code. Functions can look very different from each other, but they all follow the same general pattern. To create a function you need to go outside of the { } of the main function (That's right! main() is just a function that is run first, which is why we have to write it a little differently). The declaration looks like this:

```
int functionName(int parameter1, string parameter2){
     //code goes here
}
```

The first word is a variable type, which says what the function will return (we'll talk about that below, so just use "void" if you don't know what to put here). After that we give our function a name. This is just like variables, you can use anything as long as there's no spaces in it (it also shouldn't start with a number). After the name goes the parameters inside of a set of parens (). Each parameter is an input that we want our function to use, and we can have as many as we want, separated by commas. The examples above has 2 parameters, one int and one string. Parameters are declared just like variables, so you have to provide a type and a name, but they don't all have to match each other or be in a specific order. If you don't want to have any parameters you can just leave the space in the () blank. Finally, we have to put a set of curly braces to show what code is actually run in our function.

So, we have to go into detail about a couple parts of that explanation.
First, the function variable type. We have to say what kind of value the function will return. Just like how the main function has a "return 0;" in it, our new function has to have a return of some kind. We get to pick what kind it is though, so it can be an int, a string, a bool, or even nothing (which we call a void). If you make a function and don't return something that matches the type, then you will get an error. If you use a void then you have to just type "return;" with nothing after it (that just signals the end of the function).
Second, the parameters that we put inside the () are basically variables. The code inside of the {} can use the parameters (the int parameter1 and the string parameter2 in the examples) as if they were normal variables. They will disappear after the function returns, so don't try to use them outside the function!

So, now we need to talk about how to call (use) a function. It's very similar to the example above, but with some stuff taken out. So, if we wanted to use `functionName` from above, we would do this:
//we need to make some variables to put into the parameters:

```
int test1 = 10;
string test2 = "testString";
int result = functionName(test1, test2);
```

Whatever we return from the function will be put into `result`, which is how we use the function to get useful results.
One last note: create you function ABOVE the main function, or else your main won't recognize the name of the function because you haven't named it yet!