String Manipulation
http://www.cplusplus.com/reference/string/string/

We have used strings before, but strings can do a lot more than most of the variables we use. Strings are an object, which means that they can do more than just hold a value. Specifically, a string is an array of letters, which allows us to change parts of it without rewriting the whole thing. We do this through a process called “functions”, which we will talk about later. Right now just follow along with the examples.

String Addition:
By using the + sign we can add new symbols to the end of a string we already have, or add two strings together. For example, if first and last are both strings that hold names, then:

```
string fullname = first + " " + last;
```

Will give us a single string `fullname` that has both names, separated by a space. You can use the plus sign to add character variables to a string, too.

At:
This is the first function, so just use it like it shows. This lets us pull a single character out of the string, which we can use for whatever we want. This is very similar to how we get parts of an array:

```
char hold = fullname.at(4);
```

This would get us the 5th letter of `fullname`, and put it into the char variable `hold`.
(Please note, when checking chars, we have to use ‘e’ to specify the character e, and not “e”, which specifies a string. THEY ARE NOT THE SAME (sorry)).

Size:
This is the simplest, using this will give us a number that tells us how many characters are in the word. You don’t have to give it a number or any other input:

```
String first = "This is a very long sentence, for example";
int size = first.size();
```

This would give us a integer `size` that contains the number of characters in `first`, in this case, 32.

Substring:
This uses two numbers to pull out a subsection of the string. The first number says where we will start pulling from (***INDEXED FROM ZERO***), and the second number says how many letters to pull. Example:

```
String first = "This is a very long sentence, for example";
String second = first.substr(20, 8);
```

This example will give us the string `second` with the word “sentence” in it, because we took 8 letters after position 20.