This lab is due on Tuesday Nov 28, 11pm

Practice C++ classes in this lab (weeeeee)

**Step 1 – Basic C++ Classes (10 points)**

In this step we will be practicing your C++ skills. We are creating a basic StringList class in C++:

- **Main.cpp** – the file which will have your main test routines
- **StringList.cpp** - the cpp class code for the StringList class
- **StringList.h** the cpp header info for the class StringList

Please don’t forget to sprinkle comments liberally in your code, it will only help your grade.

You should be familiar with the idea of a String class and String list from java. Create a primitive StringList class in cpp, with the minimum following functions:

1) Constructor for taking a character pointers (c string) and inserting it into your list. You need to allocate memory and copy data into your own data representation.
2) Destructor to clean up after yourself
   - we have memory allocated, so don’t forget to delete it
3) Insert function to insert c strings into the list
4) Find function to return 0/1 (false/true)
   - if the String passed in exists in this list
5) Size function to return number of items in our list.
6) Print function to print out the contents of the StringList in some way

Summary: we want to represent a bunch of strings as a list and print them out

**Step 2 – More advanced practice (10 points)**

In addition to what we’ve done in last step, we would like to overload the

1) `<<` (left shift)
2) `+` operator
3) `!=` operator
4) `=` operator (remember its not just sizes but also items that you might need to compare

I would like to be able to code the following code:

```cpp
StringList test;  //default constructor triggered here
StringList test2;
StringList test3 = test + test2;
//Hint: some constructor is called here ?
```
if ( test != test2) {
  cout << "the lists are not equal" << "List 1 is " << test << " List 2 is " << test2 << endl;
}

cout<<"list 3 is now: " <<list3<<endl;

Hint: Look up the signature of each operator in order to overload it. For example overloading the left shift operator the following way:
friend ostream& operator<<( ostream&, const StringList &);
and define the following method:

ostream& operator <<(ostream &os,const StringList &obj)
{
   //your code goes here
   //basically loop through the items
   return os;
}

Hint: do each operator one at a time, and test before starting on next, will make the lab easier for you.

friend String operator+(const StringList& s1,const StringList& s2);

**Extra credit (5) points:**

Code a sublist function which returns true if one list is part of the other list. For example if list1 is \{A, B, C, Q, X, Z\} and list2 is \{X, C\} and list3 is \{T, Z\}

So list2 is in list1 but list3 is not i.e.

sublist(list1, list2) is true
sublist(list1, list3) is false