**CST-201**

**Project 1: MyString**

The purpose of this assignment is to practice:

* Implementing a sequential abstract data type that uses a dynamic array structure
* Analyzing and comparing algorithms for efficiency using Big-O notation

This assignment reinforces the following competency: Apply the fundamentals of mathematics in computer science disciplines.

For this assignment, you will implement a version of the string class.

Design and implement a **MyString** class defined by the following data:

* A char array reference (or pointer) for the array of characters that make up the string
* An integer **curr\_length** representing the number of characters in the string
* (C++ only) An integer **capacity** that represents the size of the array

Add the following methods to your class:

* A constructor that initializes the array to null and the **curr\_length** to 0
* A constructor that takes a String parameter and initializes the char array to the characters in the String. **curr\_length** should be appropriately initialized.
* a copy constructor that takes a **MyString** object and initializes a new **MyString** object so that it is a copy of the argument string
* a **length()** method that returns the number of characters in the string
* a private method, **ensureCapacity()**, that handles allocation of additional memory for the string
* a **toString()** method that returns a String representation of the **MyString** object (Java), or overload the insertion operator (<<) (C++).
* a **concat(MyString)** method that takes a **MyString** parameter and returns a **MyString** object that is a concatenation of the calling object and the parameter (Java) or overload the + operator (C++)
* a **.equals(MyString)** method that takes a **MyString** andter and returns true if this **MyString** and the parameter are the same (Java) or overload the **==** operator (C++)
* a **.compareTo(MyString)** method that takes a **MyString** parameter and returns as follows (Java)
  + 0 if the parameter and this **MyString** are the same
  + A negative integer if this **MyString** is alphabetically before the parameter
  + A positive integer if this **MyString** is alphabetically after the parameter

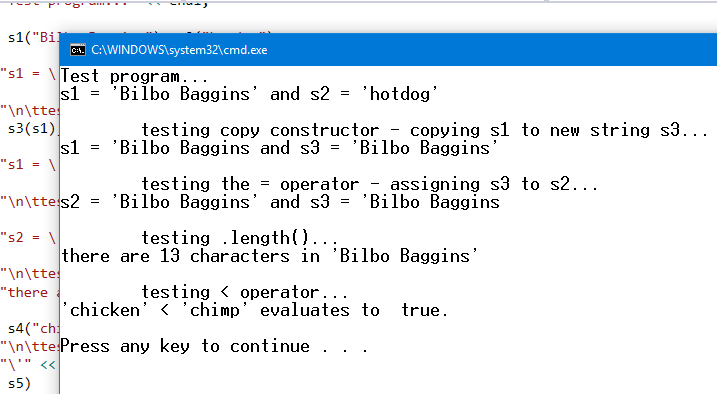
OR

Overload the < and > operators (C++)

* **.get(int)** method that takes an integer and returns the character at that index location.   
  PRE: the integer must be in range (Java) OR overload the index operator (C++)
* **.toUpper()** and **.toLower()** that return a **MyString** that is in all upper case (or lower case)
* **.substring(int)** that takes an integer and returns the substring starting at that index.
* **.substring(int n, int m)** . Return a **MyString** substring where **n** is the starting index and **m** is one past the ending index.
* **.indexOf(MyString)** and .**lastIndexOf(MyString)** that take a **MyString** parameter and return the starting index of the first (or last) occurrence of the **MyString** in the calling object. If the parameter is not found in the calling object, the method should return -1.

1. Create a driver program that tests your **MyString** class
2. Record a Loom video in which you:
   1. Offer runtime analysis for the algorithms in your **MyString** class
   2. Explain the mathematics used to calculate the number of steps required for each of your algorithms
   3. Discuss any difficulties or points of interest in your implementation of the **MyString** class

Your driver program should test ALL of the functions in your **MyString** class and should be easy to follow. Here is an example of a partial driver program that is easy to follow:



Submit the following:

* A zip file with your code
* A statement that the submitted work is your own
* A link to your Loom video

Your video must be under 4 minutes in length.