COP3330 Programming Assignment 7

MyString Class

Objectives

* Practice OOP and working with c-strings, dynamic memory allocation, overloading various operators, and copy and assignment operators.

Project details

You will implement a MyString class that stores and performs various operations with strings. In addition, you must provide a driver that tests each function of the MyString class. The MyString class overcomes some of the problems in c-string such as the unchecked [] operation, the lack of comparison operators, etc.

A complete header file, a skeleton implementation file and an empty driver file have been provided for you :

[MyString.h](file:///C%3A%5CUsers%5Csurfing.root-PC%5CApplication%20Data%5CSSH%5Ctemp%5Cgiven%5CMyString.h)
[MyString.cpp](file:///C%3A%5CUsers%5Csurfing.root-PC%5CApplication%20Data%5CSSH%5Ctemp%5Cgiven%5CMyString.cpp)
[driver.cpp](file:///C%3A%5CUsers%5Csurfing.root-PC%5CApplication%20Data%5CSSH%5Ctemp%5Cgiven%5Cdriver.cpp)

You should not change MyString.h in any way. Submissions will be tested using the MyString.h above.

The skeleton implementation file contains detailed descriptions of each of the functions. You should implement each function as described in the comments of MyString.cpp. You may use functions in the c++ standard libraries (most, if not all, functions needed should be referenced in the function descriptions) in their implementations but should **NOT** use the String class.

In addition to the completed implementation file, you must also turn in a driver file which tests the complete functionality of their implementation. Your tests should be simple but should also extensively test the correctness of your solution. The driver file should be well commented indicating where each feature is tested and the expected output.

This project is a bit longer than previous projects. You will need to use the following steps to complete the project:

1. Look over MyString.h to get an overview of what the MyString class looks like.
2. Review the function descriptions in MyString.cpp. It is not necessary to know how to implement each function yet, just get an idea of the input/output parameters and the general purpose of the function.
3. Start filling in the function definitions:
	* Start from the top of MyString.cpp -- the functions have been ordered such that implementing them in the order will be most intuitive.
	* **After you implement each routine. Write your test case in driver.cpp and test that routine!!!** This is the best time to write your test cases as you have just read the function requirements! Your test cases (driver.cpp) will count as 25% of this project, if you follow any direction, make it this one ;-)
	* Test your routine, fix bugs. Go to recitation, office hours, lecture and email your TAs and/or me with pointed questions.
	* Repeat until all functions implemented

Submission

The due time for this assignment is July 10 (Wendesday), 2013. 11:59pm.

Tar all of your files for this assignment, name the tar file yourlastname\_firstinitial\_proj7.tar and submit the tar file in blackboard.

Grading policy

The program must work on linprog. O point for programs with any g++ compiler error on linprog. For each routine in MyString.cpp class that you implemented, you must have test cases that show that your implementation works. If you failed to test a routine in driver.cpp that you implemented in the MyString class, you will get 0 point for the routine that you claim to implement (untested program is worthless). There are 25% of the points for proper testing (If you test case is inappropriate, you lose these points).

* ‘g++ MyString.cpp driver.cpp’ has no compiler error (20 points)
* Tar file name, comments in program, etc (5 points)
* Constructor from c-string and destructor (20 points)
* Resize function (5 points)
* The [] operator (two versions) (5 points)
* Assignment operator and copy constructor (10 points)
* + and += operator (5 points)
* << operator (5 points)
* >> operator (5 points)
* ++, -- operator (5 points)
* StrCpy, ReversedStrCpy(5 points)
* Comparison operators (10 points)

Hints

1. Start the project as soon as possible. This project is longer than earlier ones.