Logan Leavitt

April 3, 2019

CS 202.1103

Project 7

The source code for this project is contained in files *proj7.cpp*, *MyString.cpp*, *MyString.h*, and the *makefile* used to compile the project. Running "make" will compile an executable named *proj7* and running "make clean" will remove the executable file and the object files. The file *proj7.cpp* is a test driver designed to test the required MyString class. The sections of code are labeled 1-12 to indicate which public function each section is testing. *MyString.h* is the header file for the MyString class and contains the declarations of all required members and functions. *MyString.cpp* implements all of these functions. I did use the cstring library to implement some features, like using strcpy to copy strings to m_buffer and strcat to implement the operator+ overload. Dynamic memory allocation works as expected, buffer_deallocate will free memory pointed to by m_buffer and buffer_allocate will allocate memory to m_buffer (or deallocate and then reallocate if there is already memory assigned to m_buffer). I also implemented the allocation using a try and catch block. If the memory allocation throws a std::bad_alloc exception, "std::bad_alloc thrown in buffer_allocate" will be printed to the the console. However, if buffer_allocate is unable to allocate memory, the program will most likely result in a segmentation fault because the MyString class will try to access memory that failed to be allocated. This is something I would most likely try to fix given more time with the project. Otherwise, all the constructors successfully instantiate the MyString object and the Destructor frees the memory stored at m_buffer. All of the operator overloads work as required.