Output from driver code:

Testing Default ctor //sets string pointer to null and sets size to 0
Testing Parametrized ctor //allocates memory for string, updates size, then copies string to pointer
Testing Copy ctor //sets members to default values, then uses operator= to copy (discussed later)
Testing dtor //deletes m_buffer
Testing size() //returns the amount of characters allocated for the string
21
Testing length() //returns all characters up to but excluding the null
20
Testing c_str() //returns m_buffer as a const c-string
C-String equivalent successfully obtained!
Testing operator==() //returns 1 if all characters in each string are the same
Same success
Different success
Testing operator=() //deletes data pointed to by m_buffer (if available), then allocates and copies data from 2nd object
Testing operator+ //deletes data pointed to by m_buffer (if available), concatenates the two operands into one string, then creates and returns an object created with that string
Testing operator[]() //returns a reference to the character indexed at that point
Testing operator<<() //prints out the string stored in the object
Access successful

This project was conceptually pretty simple but I ran into a couple of problems when trying to test out my code. It was especially challenging trying to set up the virtual machine and different compilers set up because nothing would work.

All of the objectives of the project were things we've already done before (creating classes, overloading operators, making all the required functions, etc.), so that wasn't really the hard part. Using the dynamic memory wasn't too bad either, however sometimes it didn't seem like it would work properly.

When trying to compile my code, it seemed to be very inconsistent in where the program would crash or segfault. Some compilers would stop at the operator+ test, some would stop at the operator=, some wouldn't even compile, and some would run without any problems. Because of this, I had trouble actually trying to verify whether my program needed fixing or not. As of right now, I am 99% sure that it works, but not because I changed anything major, just that the compilers decided to work.

I read that some seg faults could be caused by the new operator if the freestore was full, and I noticed that deleting some of the driver code would get rid of those. This is just speculation however, and I haven't really been able to prove anything. Debugging is still a new process for me, but I think I am getting the hang of it slowly.