

CSCE 221 Cover Page
Programming Assignment #1
Due **September 11** by midnight to CSNet

First Name Anders Last Name Wallace UIN 925000221

User Name awllace2 E-mail address awallace2@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more: Aggie Honor System Office

Type of sources			
People	Zack Ramirez		
Web pages (provide URL)	www.stackoverflow.com		
Printed material			
Other Sources			

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.

“On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.”

Your Name Anders Wallace

Date 9/13/17

Program Description

The objective of this program was to create a string class called `my_string` and to implement all the functions that are contained in the `<string>` class in our own `my_string` class. We implemented various functions in order to create a working string class independent of the `string.h` class.

Data Structures Description

The data structures most implemented were static and dynamic character arrays, with accessing using character pointers (`char*`). We would hold the data of our `my_strings` by allocating memory on the heap with dynamic character arrays, and then manipulating them as needed to achieve the strings we wanted.

Compilation and Execution Instructions

In order to run the program, you can simply type in the command line:

```
g++ -std=c++14 *.cpp
```

You can also run it using an attached makefile by typing the following in the command line:

```
make all
./my_string
```

Logical Exceptions

The only logical exceptions contained are in the `at()` function, which throws an `out_of_range` error if you try to access an index does not exist inside of the array of the `my_string` object. There are no other logical exceptions or bugs

C++ and OOP Features

Some features of C++ and OOP in general that were useful here was the creation of classes with objects that were able to be manipulated using public functions, so rather than creating a complex string and having to alter it in tedious ways to get what we want, we were able to create it as an object and then perform various actions on it in a simple and easy way (such as `v1 = v2`, and `my_string v1("abc")`) rather than manually changing things in a dynamic array and so forth.

Testing Results

```
Testing my_string class:
v1 = first second
v1 size = 12
v1 capacity = 26 //I doubled size differently, so I have a different value than expected
v1 as [] characters: f i r s t   s e c o n d v1 as at() characters:
f i r s t   s e c o n d
v4 = abcd v4
size = 4 v4
capacity = 4
is v4 empty: false
v5 = first second
v5.insert(5, "ly") and v5.insert(14, "ly"):
v5 = first second //I did not do this function so the "ly" was not added
Enter a string:
```

abc

v6 = abc v6 + " " + v2 = abcfirst //this used the insert again, so I do not have the space between abc

v6 + last char of v6 = abcc