Programming Assignment 1 Report
CSCE 221

1. Program description and purpose of the assignment: The purpose of this assignment is to design and implement our own C++ string class, called my_string with all the appropriate member functions.

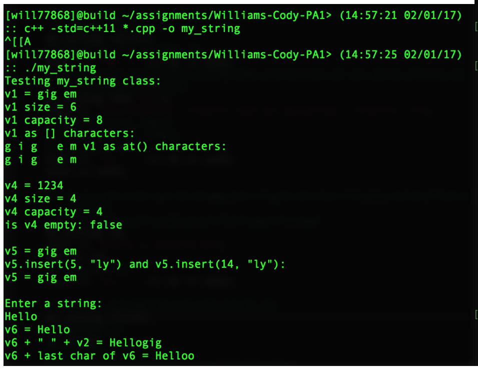
2. Data structures description: The data structure of primary focus in this assignment is the C++ array.

3. Instructions to compile and run your program including input and output specifications: In order to compile and run this program, one must use the following steps. 1) Place main.cpp, my_ctring.cpp, and my_string.h in the appropriate file on the unix server. 2) Navigate to the file where main.cpp, my_ctring.cpp, and my_string.h are located in the terminal. 3) Compile the files using the command c++ -std=c++11 *.cpp -o my_string in the terminal. 4) Run the program using command c++ -std=c++11 *.cpp -o my_string in the terminal. 5) The program will then execute and ask the user to input a string. Input a string and press enter. 6) The program will finish executing and terminate.

4. Logical exceptions (and possible bug descriptions): 1)An out_of_range exception will be thrown if the user tries to create a my_string with a negative capacity or a capacity greater than the size. ex. my_string(-3). 2)Because of the substantial use of dynamic memory allocation, there exists a possibility for memory leaks or loss of memory.

5. C++ object oriented or generic programming features, including C++11 features:The object oriented programming features included in this assignment are overloading (for the operators and constructors), utilization of the C++ standard library (when including <iostream>) , use of nullptr, and the initilization of data members in constructor

6. Testing Results:

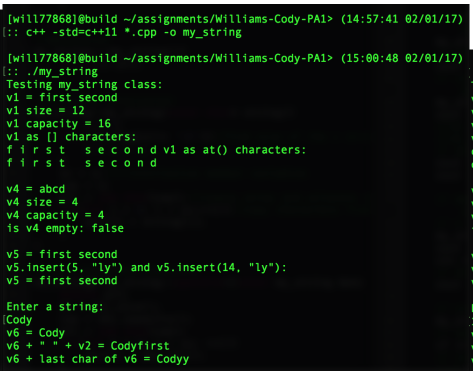


```
[will77868]@build ~/assignments/Williams-Cody-PA1> (14:57:21 02/01/17)
:: c++ -std=c++11 *.cpp -o my_string
^[[A
[will77868]@build ~/assignments/Williams-Cody-PA1> (14:57:25 02/01/17)
:: ./my_string
Testing my_string class:
v1 = gig em
v1 size = 6
v1 capacity = 8
v1 as [] characters:
g i g   e m v1 as at() characters:
g i g   e m

v4 = 1234
v4 size = 4
v4 capacity = 4
is v4 empty: false

v5 = gig em
v5.insert(5, "ly") and v5.insert(14, "ly"):
v5 = gig em

Enter a string:
Hello
v6 = Hello
v6 + " " + v2 = Hellogig
v6 + last char of v6 = Helloo
```



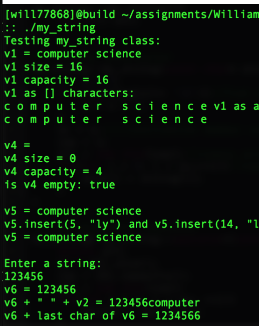
```
[will77868]@build ~/assignments/Williams-Cody-PA1> (14:57:41 02/01/17)
:: c++ -std=c++11 *.cpp -o my_string

[will77868]@build ~/assignments/Williams-Cody-PA1> (15:00:48 02/01/17)
:: ./my_string
Testing my_string class:
v1 = first second
v1 size = 12
v1 capacity = 16
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

Enter a string:
Cody
v6 = Cody
v6 + " " + v2 = Codyfirst
v6 + last char of v6 = Codyy
```



```
[will77868]@build ~/assignments/William
:: ./my_string
Testing my_string class:
v1 = computer science
v1 size = 16
v1 capacity = 16
v1 as [] characters:
c o m p u t e r   s c i e n c e v1 as a
c o m p u t e r   s c i e n c e

v4 =
v4 size = 0
v4 capacity = 4
is v4 empty: true

v5 = computer science
v5.insert(5, "ly") and v5.insert(14, "l
v5 = computer science

Enter a string:
123456
v6 = 123456
v6 + " " + v2 = 123456computer
v6 + last char of v6 = 1234566
```