

# CSCE2014 Programming Foundations II

## Homework Four

By Wing Ning Li

### 1 Problem Description

To complete the implementation of the Mystring class, the declaration of which is provided in Mystring.h file (this file is similar to the Mystring.h file of Lab 6 with minor differences). The Mystring class mimics the behavior of C++ string class.

To find out the behavior of each method to be implemented, you are encouraged to experiment with the corresponding method in C++ string class or to ask questions in the lecture. You may use c-style string library functions, such as strcpy, in your implementation. **However, you SHOULD NOT use C++ string!!** The reason is that we assume such a class is not available and therefore we are going to implement it. **Vector or any class of data structures in C++ SHALL NOT be used either.**

### 2 Purpose

Understand the concepts of class, member function and member, constructors (default, copy, etc.), destructor, assignment operator, operator overloading as a member function or a general function, memory management, array, and C++ string class. *To gain experience and understanding related to how a complex and useful class such as string may be designed and implemented using basic c++ language features, data structures, and algorithms.*

### 3 Design

Our design follows the design of C++ string. The details are provided to us in the two files: Mystring.h (on blackboard) and Mystring.cpp (your enhanced version based on the one given in Lab 6). You may expand the main.cpp file of Lab6 to test each new method or function implemented and **rename the file as homework4.cpp**.

### 4 Implementation

The subsections 4.1 to 4.3 help us to understand the assignment operator and provide hints as to how to implement it (the first version where a string is assigned to another string. Note the other version is a cstring is assigned to a string). The files being referred to in the following discussion are Lab 6 files .

## 4.1 Testing using C++ string for assignment operator

1. Comment out the line `#define string Mystring` in `main.cpp`
2. Add the following before `cout << "Lab 6 ends" << endl;` statement

```
cout << "----Testing assignment operator---\n\n";
{
    string s2;
    s2=s1;
    string s2name("s2");
    check(s2,s2name);
}
check(s1,s1name);
```

We should get the following output or something similar (perhaps a larger `max_size()` value):

```
This is Lab 6
checking s1
s1 contains Hello, World!
s1 capacity() is 13
s1 length() is 13
s1 size() is 13
s1 max_size() is 1073741820
```

```
checking s1
s1 contains Hello, World!
s1 capacity() is 13
s1 length() is 13
s1 size() is 13
s1 max_size() is 1073741820
```

```
---Testing assignment operator---
```

```
checking s2
s2 contains Hello, World!
s2 capacity() is 13
s2 length() is 13
s2 size() is 13
s2 max_size() is 1073741820
```

```
checking s1
s1 contains Hello, World!
s1 capacity() is 13
s1 length() is 13
s1 size() is 13
```

```
s1 max_size() is 1073741820
```

Lab 6 ends

Press [Enter] to close the terminal ...

## 4.2 Testing the use of Mystring default assignment operator provided by the compiler

1. Uncomment the line `#define string Mystring` in `main.cpp`
2. compile and run the same program

Do we get the same output as the previous step? Do we get a nasty run time error? Please try to explain why we get a different output in your report.

## 4.3 Implement assignment operator for Mystring class

The following provides the pseudo code or hint about how to implement the assignment operator for Mystring class as well as how to adjust the current code base. You may also study the textbook (in particular page 78).

1. Uncomment the first assignment operator prototype line in `Mystring.h`
2. Uncomment the definition assignment operator in `Mystring.cpp`
3. If the object (left hand side of `=`, and `this` is the pointer to it) that invokes the assignment operator (note `=` is a member function conceptually) is the same as the argument object (right hand side of `=`, the orig argument), for example as in (where `x` is Mystring type) `x=x`, do nothing and otherwise perform the next two steps.
4. Free the space pointed to by `ptr_buffer`
5. Allocation space based on argument object (orig) and adjust values of the members (similar to copy constructor code)
6. The last statement is `return *this`
7. Compile and run the same program

We should get the following:

```
This is Lab 6
checking s1
s1 contains Hello, World!
s1 capacity() is 14
s1 length() is 13
s1 size() is 13
s1 max_size() is 1073741820
```

```

checking s1
s1 contains Hello, World!
s1 capacity() is 14
s1 length() is 13
s1 size() is 13
s1 max_size() is 1073741820

---Testing assignment operator---

checking s2
s2 contains Hello, World!
s2 capacity() is 14
s2 length() is 13
s2 size() is 13
s2 max_size() is 1073741820

checking s1
s1 contains Hello, World!
s1 capacity() is 14
s1 length() is 13
s1 size() is 13
s1 max_size() is 1073741820

Lab 6 ends
Press [Enter] to close the terminal ...

```

#### 4.4 Implement the remaining methods or operators

**Implement all the member functions, operators, as well as non-member operators, declared in `Mystring.h`, except for the iterator methods `begin` and `end`.** The implementation goes to `Mystrng.cpp`. *You should study the relationship between the methods and think about how some operation might be implemented by other operation(s).* The idea is to reuse the code by calling other methods (an example could be homework 3 set method and constructor). Note that some methods may be implemented by calling other method(s) with a bit of additional logic. You should write the code for one method at a time, debugging and testing.

## 5 Test and evaluation

In the `homework4.cpp` program, demonstrate the functionalities of each implemented function (or operator) to be basically the same as that of the corresponding function of the string class and, therefore, the correctness of the implementation. We basically perform unit testing here (similar to homework 1, homework 2, and homework 3).

## **6 Report and documentation**

A report about things observed and things learned and understood. The report should also describe the test cases used in the main program and the reasons for each test case selected (copy and paste the testing output into your report). Properly document and indent the source code. The source code must include the author name and as well as a synopsis of the file.

## **7 Homework submission**

Use Blackboard to submit the source files (Mystring.h, Mystring.cpp, homework4.cpp) and the report.