CMPT 125 - FIC 202403 - Assignment 3

Due Date: Wednesday 13th November 2024 11:55PM

Instructor: Dr. Yonas T. Weldeselassie (Ph.D.)

Read this document in its entirety and carefully before you start anything and understand it. If you have any questions, don't hesitate to email me.

Academic misconduct, plagiarism, and cheating

You are strongly recommended to avoid any academic misconduct, plagiarism, or cheating in the assignment work. During the final exam, it will be presumed that you have practised with the assignment content and as such you may be asked questions related to the assignment. It will therefore be to your advantage to work on the assignment by yourself in order not to lose much more marks in the final exam.

Problem Statement

In this assignment, you will implement a class named **CMPT125_String** that will mimic the C++ string class. The aim is for you to apply what you have learned in the course regarding

- Design of classes,
- Constructor member functions,
- Destructor member function,
- Assignment operators,
- Getter and setter member functions,
- Member operator functions,
- friend operator functions, and
- Other non-member and non-friend functions.

In order for all of you to have the same class design, that is to say the same member, friend, and non-member and non-friend function signatures; you will be provided a text file named `StarterCode.txt` that contains the class declaration, partial definition, and descriptions of all the member, friend, and non-member and non-friend functions that need to be defined. The `StarterCode.txt` file also contains a test program and its sample run output that minimally tests the functionality of the **CMPT125_String** class and the non-member and non-friend functions in order to help you test your work.

Discussion

We would like our **CMPT125_String** class to represent strings internally as cstrings. This means our class will have **ONLY ONE** member variable which is a pointer to char data type. This pointer member variable will be named **`buffer`**.

In order to avoid any memory leak, an empty **CMPT125_String** object will be represented by assigning the buffer a **`nullptr`** value. That is an empty **CMPT125_String** object will have NO memory allocated to it whatsoever (not even for a terminating null character). Of course any non-empty **CMPT125_String** object will have memory allocated for all of its printable characters and a null terminating character.

Finally, the destructor member function will be designed in such a way that after destruction a **CMPT125_String** object will have its buffer assigned a nullptr value. This will avoid any memory leak when objects go out of scope.

Restriction

- *You are NOT allowed to add any include directive into your program besides the ones already included in the starter code file.*
- *You are NOT allowed to change any given function signature. Use the function declarations exactly as they are given in the starter code file including the spelling and capitalization. If you change any function signature, you will lose all your marks.*
- *You are NOT allowed to remove or add any member function to the given class declaration.*

Requirement

You are required to define all the member, friend, and non-member and non-friend functions as described in the starter code file. You should implement your functions in such a way that the given test program runs without any syntax, linking, runtime, or semantic errors and provide the same output as the provided sample run output.

You are also required to submit an original work. Submitting any portion of work that is not original is academic misconduct and will be penalized according to the policies set out in the course outline.

Starting Your Work

In order to start your work right away, please copy and paste the given code from the **`StarterCode.txt`** file into your project.

Submission Format

You will find a submission button for Assignment 4 on Moodle under Week 9 and you are required to upload the source code of your program (that is .cpp file) containing the class declaration, class definition, the non-member and non-friend function definitions, and the test main program.

Submission Due Date and Time

The due date and time to submit your work through Moodle is **Wednesday 13th November 2024 at 11:55PM**. Moodle will not allow you to submit after this date and time.

Marking

Your program will be tested under Microsoft Visual C++ 2010 Express and you are advised to test your program on the same IDE before submitting your work. A program with any syntax or linking errors will automatically get zero mark. A program with runtime or semantic errors will lose marks depending on how severe its shortcoming is when it is executed.