Data Structure and Algorithm

Analysis--- COP3530 Program –

Module 5

Total Points: 25

In this assignment you will implement a class called "**string_class**". Place the class declaration in the file "**string_class.h"** and the class implementation in the file "**string_class.cpp"**. The class has the following characteristics"

1. A private string variable "**current_string**".
2. A **default constructor** that sets "**current_string**" to an empty string ("").
3. An **explicit-value constructor** that sets "current_string" equal to the argument that is passed to the explicit-value constructor when a **string_class** object is declared.
4. A public member Boolean function called "**palindrome**" that returns true if the current_string reads the same forward as it does backwards; otherwise it return false. For example "madam", "463364", and "ABLE WAS I ERE I SAW ELBA" are all palindromes.
5. A public member void function called "**replace_all**" that accepts two string arguments, " **old_substring**" and "**new_substring**". The function will replace each occurrence of "old_substring" with "new_substring" in "**current_string**". For example, when the function is invoked, if current_string = "aaabbacceeaaa", old_substring = "aa", and new_substring= "zzz", then after execution of the function, current_string = "zzzabbacceezzza". **Note special cases: If current_string is empty, or if old_substring is larger than current_string, or if old_substring is not located in current_string, then the value of current_string will not change. DO NOT USE THE STRING CLASS FUNCTIONS "find", "replace", or "substr".**
6. Overload **the insertion operator (<<)** as a friend function of the class with chaining to print the contents of a string_class object's "current_string".
7. You may implement other class member functions if necessary.

Call the driver to test the functionality of string_class, **"stringclass_driver.cpp"**. **You should submit the files "string_class.h", "string_class.cpp", and "stringclass_driver.cpp"** to Canvas before the due date and time.

See the sample main program below

sample_main_program(driver)_for_stringclass:

```
#include <iostream>
#include <fstream>
#include <string>
#include "string_class.h"
using namespace std;

int main()
{
	/*string_class s;

	cout << "****************************************" << endl
		<< "Test#1: tesing default constructor and overloaded operator<< with chaining\n"
		<< s << "1st blank line" << endl << s << "2nd blank line" << endl
		<< "Test#1 Ended" << endl
		<< "****************************************" << endl;
	string_class r("hello");
	cout << "****************************************" << endl
		<< "Test#2: tesing explicit-value constructor and overloaded operator<< with
chaining\n"
		<< r << endl << "1st blank line" << endl << r <<endl<< "2nd blank line " << endl
		<< "Test#2 Ended" << endl
		<< "****************************************" << endl;
	cout << "****************************************" << endl
		<< "Test#3: tesing palindrome\n"
		<< "****************************************" << endl;
	string response = "Y";
	string ss;
	while (response == "Y" || response == "y")
	{
		cout << "Enter String: ";
		getline(cin, ss);
		string_class main_string(ss);

		if (main_string.palindrome())
		{
			cout << ss << " is a palindrome\n";
		}
		else
		{
			cout << ss << " is not a palindrome\n";
		}
		cout << "Would you like to try another string? (Y or N):    ";
		getline( cin,response);
	}
	cout << "Test#3 Ended" << endl
		<< "****************************************" << endl;
	cout << "****************************************" << endl
		<< "Test#4: tesing replace_all\n"
		<< "****************************************" << endl;
	response = "y";
	string current, old_substring, new_substring;
	while (response == "Y" || response == "y")
	{
		cout << "Enter value for current_string: ";
		getline(cin, current);
		string_class current_string(current);
		cout << "Enter old_substring: ";
		getline(cin, old_substring);
		cout << "Enter new_substring: ";
		getline(cin, new_substring);
		cout << "Current Value in Current string = " << current << endl;
		current_string.replace_all(old_substring,  new_substring);
```

```
            cout << "New value in Current String = " << current_string << endl;
            cout<< endl;
              cout << "Would you like to try another string? (Y or N):    ";
              getline(cin, response);
              cout << endl;
        }
        cout << "*************************************" << endl
              << "End Test#4: tesing replace\n"
              << "*************************************" << endl;*/  return 0;
}
```