Exp-01: C++ string and different string operations

a) Objectives:

- Understand how to take input and print strings using C++ string class.
- Learn essential string operations using C++ string such as copy, concatenation, and comparison.

b) Prerequisites:

- Familiarity with data types and arrays
- Basic knowledge of loops and functions
- Understanding of standard input/output in C++

c) Theory:

C++ supports two ways of handling strings:

• C-style strings using arrays of characters ending with null terminator '\0'.

• C++ string class, part of the Standard Template Library (STL), offering powerful and flexible operations.

d) Lab Tasks

Task 1: String Input and Output

Description: Take input from the user and print it using both C-style char arrays and C++ string class. Demonstrate input/output with and without spaces.

Sample Input:
Hello
This is a line

Sample Output:
char array (no spaces): Hello
char array (with spaces): This is a line
string object: This is a line

Task 2: C++ String Operations

Implement string operations using C++ string, including:

- Copy one string to another
- Concatenate two strings

- Get a substring
- Replace part of a string
- Erase characters from a string
- Swap two strings

Sample Input:
str1 = Hello, str2 = World

Sample Output:
Copied: Str3 = Hello
Concatenated: HelloWorld
Substring (0 to 4): Hell
Replaced: Wxrld
Erased: Wld
Swapped: str1 = World, str2 = Hello

Task 3: Reverse a given string.
Description: Reverse a string.

Sample Input: Coding
Sample Output:
Reversed: gnidoC

Task 4: Uppercase to lowercase and lowercase to uppercase conversion.
Description: Convert all the uppercase character to lowercase and lowercase character to uppercase in a given string.

Sample Input: WizArD WorLD
Sample Output:
Reversed: wIZaRd wORld

Task 5: Check Palindrome
Description: Check if a given string is a palindrome (reads the same forward and backward).

Sample Input: madam

Sample Output: Palindrome

Task 6: Substring and Search
Description: Extract a substring from a given string and search for a substring within it.

Sample Input:
Main string: programming
Substring to search: gram

Sample Output:
Substring: gram
Found at index: 3

e) Discussion

- char arrays are suitable for lower-level string operations but require manual handling.
- The C++ string class is flexible and provides robust built-in functionality.
- Using STL string functions improves code readability and reduces errors.

f) Homework Tasks

1. Remove all vowels from a given string.
2. Reverse a string and convert all lowercase characters to uppercase and uppercase to lowercase.
3. Count the frequency of each character in a string.
4. Write a C++ program that counts the number of unique characters in a string.
5. Write a C++ program that removes a specific word from a given string. Return the updated string.

   Test Data:

   ("Exercises Practice Solution", "Solution") -> "Exercises Practice"
   ("Exercises Practice Solution", "Practice ") -> "Exercises Solution"
   ("Exercises Practice Solution", " Solution") -> " Practice Solution"