**National University of Computer & Emerging Sciences (NUCES) Islamabad,**

Department of Computer Science

**DATA STRUCTURES – Spring 2023**

**LAB 10**

**Learning Outcomes**

In this laboratory, you will implement infix to pre-fix and post-fix using Stacks.

For this lab, you may use the built-in stack library to attempt the following questions.

#include <stack>

The functions associated with stack are:   
[empty()](https://www.geeksforgeeks.org/stack-empty-and-stack-size-in-c-stl/) – Returns whether the stack is empty  
[size()](https://www.geeksforgeeks.org/stack-empty-and-stack-size-in-c-stl/) – Returns the size of the stack  
[top()](https://www.geeksforgeeks.org/stack-top-c-stl/) – Returns a reference to the top most element of the stack  
[push(g)](https://www.geeksforgeeks.org/stack-push-and-pop-in-c-stl/) – Adds the element ‘g’ at the top of the stack  
[pop()](https://www.geeksforgeeks.org/stack-push-and-pop-in-c-stl/) – Deletes the most recent entered element of the stack

Question 1:

Write a C++ program to check if a given infix expression is balanced using stacks.

Question 2:

Implement a C++ function to convert an infix expression to a prefix expression using stacks.