National University of Computer and Emerging Sciences

Laboratory Manual

for

Data Structures Lab

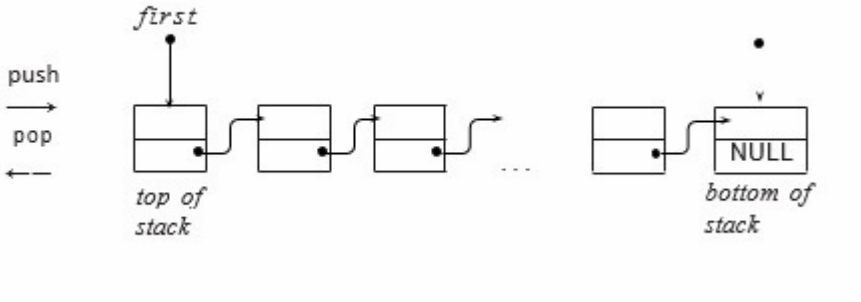
Course Instructor	Mr. Muhammad Naveed
Lab Instructor	Mr. Durraiz Waseem
Section	BDS-3B
Date	Sept 30, 2025
Semester	Fall 2025

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives: Stack and Queue implementation using linked list

Task 1: Implement a template-based stack using a singly linked list (you can reuse its code from Lab#3)



The required member methods are:

a. **int size()**: returns the count of total element stored in the stack.
b. **bool isEmpty()**: returns true if the stack is empty else false.
c. **bool top(T&)**: returns, but does not delete, the topmost element from the stack via the parameter passed by reference. It returns false via a return statement if there is no element in the stack, else it returns true and assigns the top most element to the parameter passed by reference.
d. **void pop()**: deletes the top most element from the stack. If there is no element, return some error.
e. **push(T const& e)**: pushes the element "e" on top of the stack
f. **void display()**: print all the elements of stack in order. (From **bottom** i.e. first element pushed in stack to **top** i.e. last element pushed in stack)

Task 2

Write a function that converts an expression from infix form to post-fix form.
Input: A + B * C + D
Output: ABC*+D+

Task 3

Determine if a given string is a palindrome using a stack.
Input: sasas
Output: given string is a palindrome

Task 4: Implementing a Queue

Implement a template-based queue using **Linked List**. The required member methods are:

a) **void enqueue()**: Adds an element to queue
b) **void dequeue()**: Removes an element from queue
c) **bool isFull():** return true if queue is full else false.
d) **int size()**: returns the count of total element stored in the stack.
e) **bool isEmpty()**: returns true if the stack is empty else false.
f) **int front()**: returns the element on Front of queue
g) **int rear()**: return the element on Rear of queue

Task 5: bool isPalindrome()

Global function (outside the class) checks if the elements in the queue form a palindrome. A palindrome is a sequence that reads the same forwards as backward. You should ignore the wrap-around when checking for palindromes. For example, if the queue contains [1, 2, 3, 2, 1], it is considered a palindrome because [1, 2, 3, 2, 1] reads the same backward.

Task 6: Reverse a Queue

Write a function **reverseQue**() to reverse the elements of a queue. Hint: You can implement this using a stack.