National University of Computer and Emerging Sciences

Laboratory Manual

for

Data Structures Lab

Course Instructor	Dr. Amna Khan
Lab Instructor(s)	Asad Ullah Maham Naeem
Section	CS-F
Semester	Fall 2020

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

In this lab, students will practice:
Stack Implementation using Singly Linked list
The applications of Stacks

1. Implement a template-based stack using a Singly Linked list. The required member methods are:

   int size(): returns total element stored in the stack
   bool isEmpty(): returns true if the stack is empty else false.
   T const& top(): returns, but does not delete, the topmost element from the stack if there is no element, return some error.
   void pop(): deletes the top most element from the stack. If there is no element, return some error.
   push(T const& e): pushes the element "e" on top of the stack if there is some space available. Otherwise it returns some error.

   Below functions are global functions.

2. Reverse a given string using a Stack.
   void Reverse(stack<T> ,string &s)

   Sample input: "computer"
   Sammple output: "retupmoc"

3. Given an expression containing opening and closing braces, brackets, and parentheses; implement a function "isBalanced" to check whether the given expression is a balanced expression or not, using your stack implementation. For example, {[{}{}]}[()], {{}{}}, and []{}() are balanced expressions, but {()}[) and {(} are not balanced. In your main function test your function using the given examples.
   `bool isBalanced(stack<T>,string exp)`

   Sample input: {[{}{}]}[()] Or {{}{}}, or []{}()
   Sample answer: "balanced"