# 

# Department of Software Engineering

**CS 250: Data Structures and Algorithms**

**Class: BESE-7AB**

**Lab 04: Stacks**

**CLO1: Understand the fundamentals of data structures and algorithms**

**Date: October 6th, 2017**

**Time: 09:00 am -12:00pm, 2:50pm – 5:00pm**

# Instructor: Dr. Muhammad Shahzad

# Lab 4: Implementation of Stack ADT

**Introduction**

This lab consists of stacks implementation and some of its applications.

**Objectives**

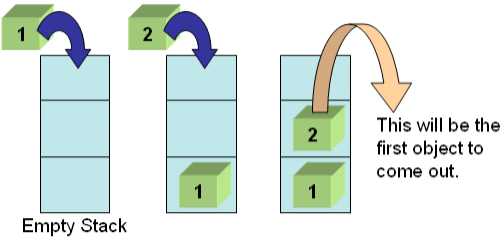
Objective of this lab is to enable students to build stack ADT using linked list and arrays, perform the following tasks on it and analyze the performance of each implementation.

**Tools/Software Requirement**

Visual Studio c++

**Description**

A stack meant to mimic the information storage and retrieval in LIFO (Last In First Out) order.



**Stack Operations**

1. void Push(element) – pushes an element on the top of stack
2. element Pop() – removes and display the element on the top of stack
3. boolisEmpty() – checks if the stack is empty or not
4. boolisFull() – checks if the stack is full or not
5. void Clear() – release the memory allocated by stack
6. void Peek() – display the contents of top element of stack

**Lab Tasks**

You are required to upload the lab tasks on LMS and the name of that tasks must be in this format YourFullName\_reg#.cpp

Remember to comment your code properly. Inappropriate or no comment will results in deduction of marks.

**Task 1:**

The idea is rather simple: You keep a Stack of braces, and every time you encounter an open brace, you push it into your stack. Every time you encounter a close brace, you pop the top element from your stack. At the end, you check your stack for being empty. If so, indeed your input string contained balanced braces. Otherwise, it didn't.

**Expected Input**

1. 1 + 2 \* (3 / 4)
2. 1 + 2 \* [3 \* 3 + {4 – 5 (6 (7/8/9) + 10) – 11 + (12\*8)] + 14
3. 1 + 2 \* [3 \* 3 + {4 – 5 (6 (7/8/9) + 10)} – 11 + (12\*8) / {13 +13}] + 14

Your program will determine whether the open brackets (the square brackets, curly braces and the parentheses) are closed in the correct order.

**Expected Output**

1. This expression is correct.
2. This expression is NOT correct. Error at character # 10. ‘{‘- not closed.
3. This expression is correct.

Your program should be able to take generic input expression from user

Solve the above problem using an **array based stack**.

**Deliverables**

Students are required to upload the lab on LMS before the deadline.