# University of Central Punjab

Faculty of CS ad IT

**Course Code:** CSAL1213 **Course Title:** Discrete Structures
**Assessment Tool:** Assignment 1 **Term:** Spring 2023
**Teacher:** Rabia Arshad **Date of Assignment:** 8th April, 2023
**Submission Date:** 13th April, 2023 **Section:** C8

**Instructions:**

- Submission after the mentioned date will not be accepted. Only portal generated submissions will be accepted.
- You must upload a folder containing your code file, screen shot of run time, as well as text file (notepad) of code.
- Please write the names of file with as Question No, while uploading.
- For Question 3, hand written assignment would be prepared and upload as a combined PDF file.
- For further queries you can contact at rabia.arshad@ucp.edu.pk

***Note: Do not use #include <stack> in your code. Use template, classes (base class, child class) as instructed in the lecture to implement the code.***

**Question 1:**

Implement a stack structure to check the balanced parentheses in an expression.
sample input: (a+b+c)
output: balanced expression.

sample input: (a+b
output: not balanced

**Question 2:**

The compiler scans the expression either from left to right or from right to left.
Consider the expression: a + b * c + d
The compiler first scans the expression to evaluate the expression "b * c", then again scan the expression to add "a" to it. The result is then added to "d" after another scan. The repeated scanning makes it very in efficient. It is better to convert the expression to postfix (or prefix) form before evaluation. The corresponding expression in postfix form is: "a b c*+d+". The postfix expressions can be evaluated easily using a stack.
You have to implement the following functionality using stack which takes input of fully parenthesized infix expressions and convert it to postfix form.

**Question 3:**

a) Convert the expression into infix and prefix notation. 4 2 − 3 / 5 3 4 + ∗ 2 / + 7 +
b) Convert the expression into postfix and prefix notation. A + ((B + C) * (E - F) - G) / (H - I)
c) Convert the expression into postfix and infix notation. + + * + A B C / D E / * F G H