

DATA STRUCTURES — FALL 2023

LAB 07

Learning Outcomes

In this lab,
You will implement the Stack using linked list.

Task 1: Implement Stack ADT

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. bool isEmpty () – checks if the stack is empty or not
4. bool isFull () – checks if the stack is full or not
5. void Clear () – release the memory allocated by stack
6. void Peak () – display the contents of the top element of stack
7. Constructor ()
8. Destructor ()
9. Print Stack ()

Task 2:

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 e.g., error at character # 10. '{' - not closed.
3. This expression is correct.

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