Lab 6

This lab is to be completed individually

This lab is for you to understand Stack data structure implementation and its application

Part -1

You need to implement stack with LinkedList.

Use "emp.txt" file from lab4.
Read line in file and create employee object and push them on stack.
You will add items to head of the LinkedList. While removing you will remove item from head of the LinkedList.

Do following operations on Stack-
Push/Pop/Peek
Push employee objects (Created by reading file) on stack.

Program input

File "emp.txt"

Program output

Print top element from stack.
Then Pop two items from stack.
Do peek operation on stack and print now top element from stack.

Note: You can use in-built LinkedList class (Java) or you can create your own LinkedList class. Creating own LinkedList class is optional.

Part-2

Infix to postfix evaluation: Using the Linked List version of the Stack class developed in the previous part, evaluate the following expressions. Note that you have to first change these infix expressions to postfix expressions. Once you have a postfix expression, evaluate it using the Stack class to get the result.

Here is some input data for you to test your program:
(1) 1 + 3 * 8
(2) 8 - 2 + 8 / 4 + 6 - 1 - 6 / 2
(3) 8 - 3 - 4 * 6 + 3

Print postfix and evaluation of all above three inputs.

Note: One document giving infix-postfix evaluation is uploaded on blackboard for your reference.