

Statement Purpose:

Purpose of this Lab is to familiarize the students with the use of stack data structure in writing simple Java programs. Another aim is to teach the students how to implement stack data structure using arrays and linked list. The students are given small tasks related to stack which they complete during the lab session under the supervision of the lab instructor. This helps them understand the concepts well which they learnt in their lectures.

Activity Outcomes:

The students will learn how to write methods related to some actions performed on stacks. They will understand how to write simple methods to

- Evaluate postfix expression using stack
- Convert infix expression into its equivalent postfix expression

Theory Review (20 Minutes):**(i) Stack Implementation using array: (10 Minutes)**

Review of Stack code using array (all related files are uploaded on Blackboard)

(ii) Stack Implementation using linked list: (10 Minutes)

Review of Stack code using linked list (all related files are uploaded on Blackboard)



LAB EXERCISES: (60 Minutes)

1. Add a method named EvalPostfix() in the above program (implemented with array) which evaluates any postfix expression using stack.

Sample Run 1:

```
Output - LabActivities (run) ☒
run:
-----|
-----|      Stack - Array (Menu)      |-----|
-----|-----|
|  1. Push an item into the stack      |
|  2. Pop (and print) an item from the stack |
|  3. Peek (look at) the top item in the stack |
|  4. Search for an item in the stack |
|  5. Print all nodes in the stack |
|  6. Evaluate Postfix expression |
|  7. Convert Infix to Postfix |
|  8. Quit |
-----|-----|

> Please enter your choice: 6
Enter the Postfix expression you wish to evaluate (type on one line and use spaces between all terms):
Example: 7 16 * 5 + 16 * 3 + 16 * 1 +
Please enter the postfix expression: 5 4 * 3 + 2 /
> You entered the Postfix expression: 5 4 * 3 + 2 /
> This evaluates to 11
```



Sample Run 2:

```

Output - LabActivities (run)
-----
Stack - Array (Menu)
-----
1. Push an item into the stack
2. Pop (and print) an item from the stack
3. Peek (look at) the top item in the stack
4. Search for an item in the stack
5. Print all nodes in the stack
6. Evaluate Postfix expression
7. Convert Infix to Postfix
8. Quit

> Please enter your choice: 6
Enter the Postfix expression you wish to evaluate (type on one line and use spaces between all terms):
Example: 7 16 * 5 + 16 * 3 + 16 * 1 +
Please enter the postfix expression: 7 16 * 5 + 16 * 3 + 16 * 1 +
> You entered the Postfix expression: 7 16 * 5 + 16 * 3 + 16 * 1 +
> This evaluates to 30001

```

2. Add a method named Infix2Postfix() in the above program (implemented with array) which converts any infix expression into its equivalent postfix expression.

Sample Run 1:

```

Output - LabActivities (run)
-----
run:
-----
Stack - Array (Menu)
-----
1. Push an item into the stack
2. Pop (and print) an item from the stack
3. Peek (look at) the top item in the stack
4. Search for an item in the stack
5. Print all nodes in the stack
6. Evaluate Postfix expression
7. Convert Infix to Postfix
8. Quit

> Please enter your choice: 7
Enter the Infix expression you wish to convert (type on one line and use spaces between all terms):
Example: 7 * 16 + 5 + 16 * 3 + 16 * 2
Please enter the Infix expression: 7 * 6 + 3
> You entered the Infix expression: 7 * 6 + 3
> Which converts to the following Postfix expression: Postfix Expression 76*3+

```



