Advanced Data Structures Lab
ACADEMIC YEAR 2025-2026

MCA Semester 1	Subject : Advanced Data Structures Lab
Name : Mukund Gangurde	Unit 3 : Stack Topic : Application of Stack 1) Evaluation of a postfix expression 2) Balancing of parenthesis
Roll No. : MCA2511	Date : 10-10-2025

1. Evaluation of a postfix expression

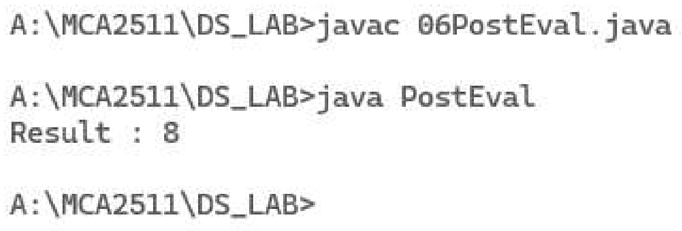
Code:
06PostEval.java

```
import java.util.*;

class PostEval
{
	public static void main(String[] args)
	{
		String expr = "23*5+62/-";

		int result = CalcPostfix(expr);
		System.out.println("Result : " + result);
	}//end of psvm

	public static int CalcPostfix(String ex)
	{
		int[] stack = new int[ex.length()];	//Stack
		int tos = -1;				//TOS

		//Loop through the ex
		for(int i=0; i<ex.length();i++)
		{
			char ch = ex.charAt(i);

			//If ch is a number - push it on the stack
			if(Character.isDigit(ch))
			{
				tos++;
				stack[tos] = ch -'0';
			}
			else if (ch=='+' || ch=='-' || ch=='*' || ch=='/')
			{
				int x2 = stack[tos--];
				int x1 = stack[tos--];
```

```
                    int res = 0;

                    switch(ch)
                    {
                         case '+':
                              res = x1 + x2;
                              break;
                         case '-':
                              res = x1 - x2;
                              break;
                         case '*':
                              res = x1 * x2;
                              break;
                         case '/':
                              res = x1 / x2;
                              break;
                    }//end of switch
                    //Push res back on the stack
                    tos++;
                    stack[tos] = res;
               }//end of if else
          }//end of for loop i
          return stack[tos];
     }//end of CalcPostfix
}//end of PostEval
```

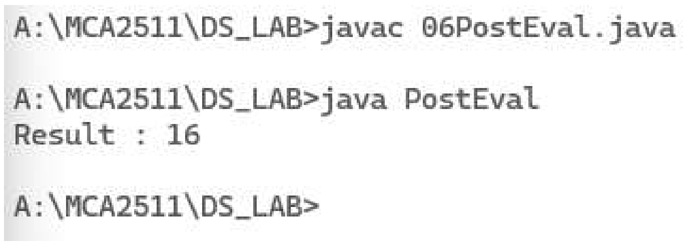
Output:

For, String expr = "23*5+62/-";



```
A:\MCA2511\DS_LAB>javac 06PostEval.java

A:\MCA2511\DS_LAB>java PostEval
Result : 8

A:\MCA2511\DS_LAB>
```

For, String expr = "53*5+82/-";



```
A:\MCA2511\DS_LAB>javac 06PostEval.java

A:\MCA2511\DS_LAB>java PostEval
Result : 16

A:\MCA2511\DS_LAB>
```

2. Balancing of parenthesis

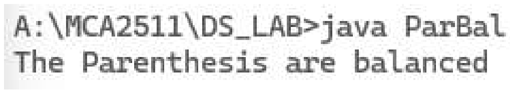
Code:
061ParBal.java

```
import java.util.*;

class ParBal
{
	public static void main(String[] args)
	{
		String expr = "((a+b)*(c+d))";

		if (isBalanced(expr))
		{
			System.out.println("The Parenthesis are balanced");
		}
		else
		{
			System.out.println("The Parenthesis are not balanced");
		}
	}// end of psvm

	public static boolean isBalanced(String ex)
	{
		char[] stack = new char[ex.length()];
		int tos = -1;

		//Scan Expression
		for (int i=0; i<ex.length(); i++)
		{
			char ch = ex.charAt(i);

			//Open parenthesis push on the stack
			if (ch=='(')
			{
				stack[++tos] = ch;
			}
			else if(ch==')') // Close parenthesis pop
			{
				if (tos==-1)
				{
					//No matching open parenthesis
					return false;
				}
				tos--;	//Pop from the stack
			}
```

```
                } /// end of for i

                return tos==-1;                    // Return if stack is empty false otherwise
        }// end of isBalanced

}// end of ParBal
```

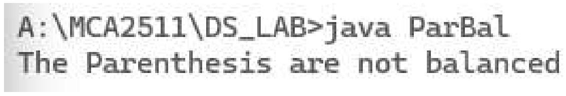
Output:

String expr = "((a+b)*(c+d))";



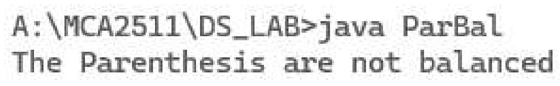
```
A:\MCA2511\DS_LAB>java ParBal
The Parenthesis are balanced
```

String expr = "(a+b)*(c+d))";



```
A:\MCA2511\DS_LAB>java ParBal
The Parenthesis are not balanced
```

String expr = "((a+b)*(c+d)";



```
A:\MCA2511\DS_LAB>java ParBal
The Parenthesis are not balanced
```