MET
Bhujbal Knowledge City
MET INSTITUTE OF COMPUTER SCIENCE

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 3) Conversion of Infix to Postfix
Roll No. : MCA2511	Date : 13-10-2025

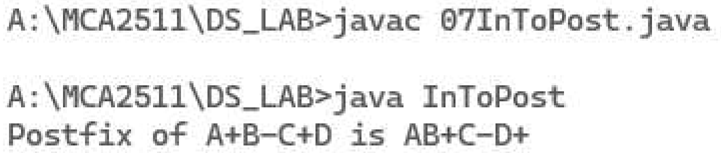
1. Using a Stack Data Structure, convert a given infix expression to postfix.

Code:

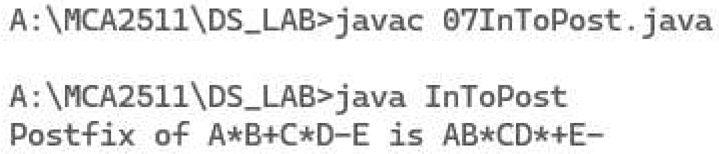
07InToPost.java

```
import java.util.*;

class InToPost
{
        public static void main(String[] args)
        {
                String expr = "A+B-C+D";
                String postfix = infixToPostfix(expr);
                System.out.println("Postfix of A+B-C+D is " + postfix);
        }//end of psvm

        //convert infix to postfix
        static String infixToPostfix(String ex)
        {
                char[] stack = new char[ex.length()];
                int tos = -1;
                StringBuilder res = new StringBuilder();

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

                        //if character is an operand - add it to res
                        if(Character.isLetterOrDigit(ch))
                        {
                                res.append(ch);
                        }
                        else if(ch=='+' || ch=='-' || ch=='*' || ch=='/') //is an operand
                        {
                                while(tos!=-1 && precedence(stack[tos]) >= precedence(ch))
                                {
                                        res.append(stack[tos--]);
                                }
                stack[++tos] = ch;
            }
        }//end of for i

        //Pop the stack and add to the res
        while(tos!= -1)
        {
            res.append(stack[tos--]);
        }
        return res.toString();
    }//end of infixToPostfix

    //Precedence
    static int precedence(char op)
    {
        switch(op)
        {
            case '+':
            case '-':
                return 1;
            case '*':
            case '/':
                return 2;
            default:
                return -1;
        }
    }//end of precedence
}//end of class InToPost
```

Output:



```
A:\MCA2511\DS_LAB>javac 07InToPost.java

A:\MCA2511\DS_LAB>java InToPost
Postfix of A+B-C+D is AB+C-D+
```



```
A:\MCA2511\DS_LAB>javac 07InToPost.java

A:\MCA2511\DS_LAB>java InToPost
Postfix of A*B+C*D-E is AB*CD*+E-
```