

```

1  import java.util.Stack;
2  import java.util.Scanner;
3
4  public class InfixToPostfix
5  {
6      static int Prec(char ch)
7      {
8          switch (ch)
9          {
10             case '+':
11             case '-':
12                 return 1;
13
14             case '*':
15             case '/':
16                 return 2;
17
18             case '^':
19                 return 3;
20             }
21             return -1;
22         }
23
24
25     static String infixToPostfix(String exp)
26     {
27
28         String result = new String("");
29
30         Stack<Character> stack = new Stack<Character>();
31
32         for (int i = 0; i<exp.length(); ++i)
33         {
34             char c = exp.charAt(i);
35
36             if (Character.isLetterOrDigit(c))
37                 result += c;
38
39             else if (c == '(')
40                 stack.push(c);
41
42             else if (c == ')')
43             {
44                 while (!stack.isEmpty() && stack.peek() != '(')
45                     result += stack.pop();
46                 stack.pop();
47                 /*if (!stack.isEmpty() && stack.peek() != '(')
48                     return "Invalid Expression"; // invalid expression
49                 else
50                     stack.pop(); */
51             }
52             else
53             {
54                 while (!stack.isEmpty() && Prec(c) <= Prec(stack.peek()))
55                     result += stack.pop();
56                 stack.push(c);
57             }
58         }
59
60         while (!stack.isEmpty())
61             result += stack.pop();
62
63         return result;
64     }
65
66
67     public static void main(String[] args)
68     {
69         Scanner in=new Scanner(System.in);

```

```
70         String exp = in.nextLine();
71         System.out.println(infixToPostfix(exp));
72     }
73 }
```