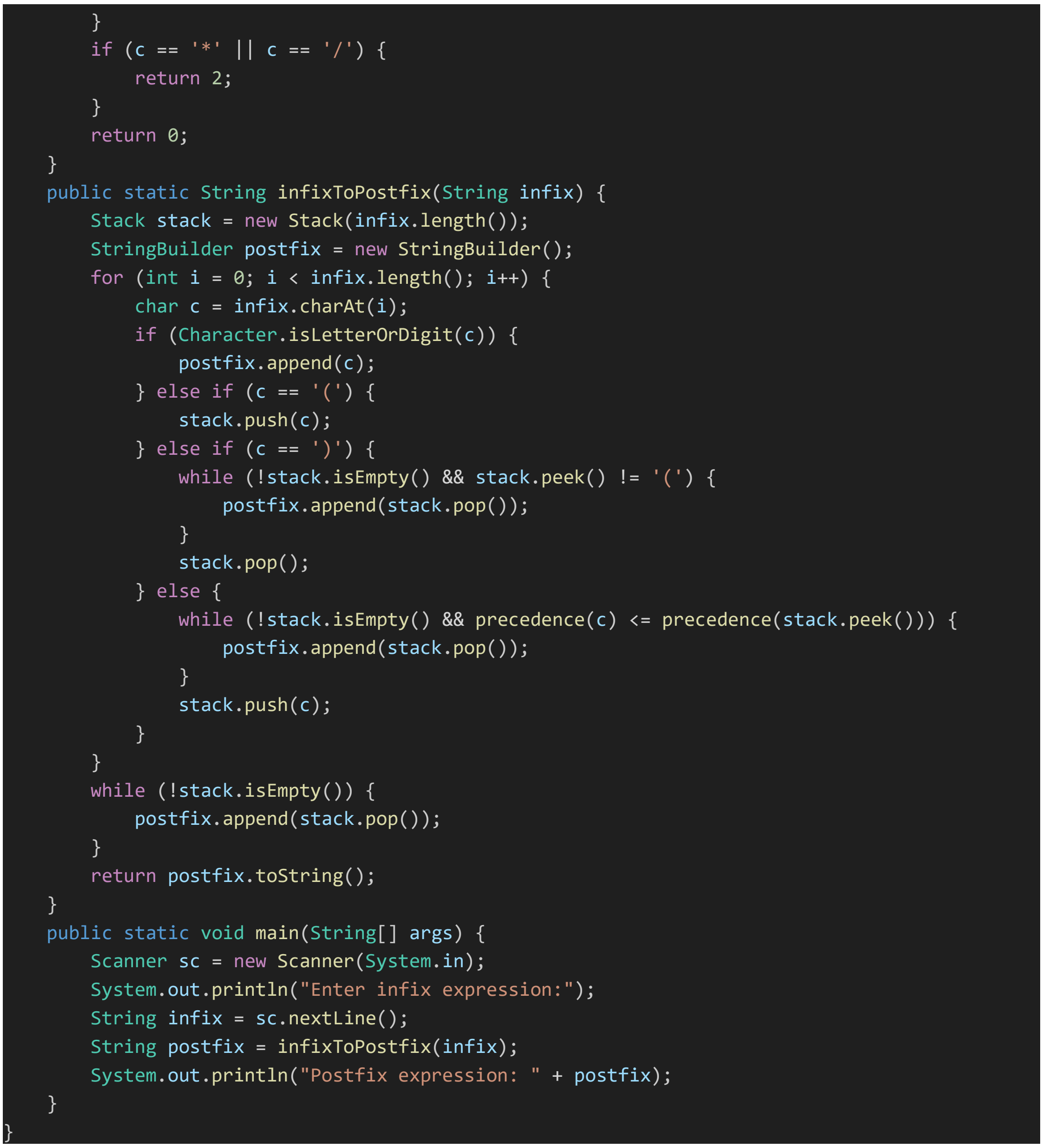
Name : Gajanan Buchake
Batch : E1
Prn : 202301100018

write a java program to convert infix expression to postfix expresssion. Implement the class stack with the methods push, pop and topofstack. Use the class stack in the conversion.

Code -

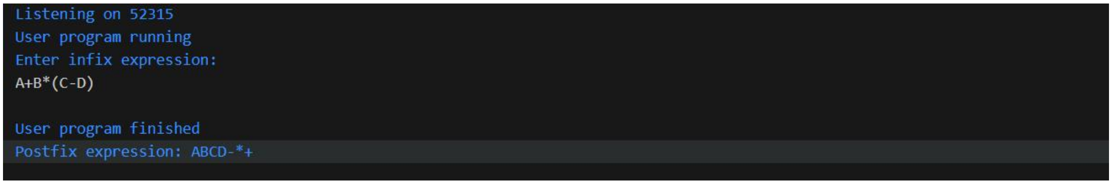
```
import java.util.Scanner;

class Stack {
    private char[] stack;
    private int top;
    private int size;
    public Stack(int size) {
        this.size = size;
        stack = new char[size];
        top = -1;
    }
    public void push(char c) {
        if (top < size - 1) {
            stack[++top] = c;
        }
    }
    public char pop() {
        if (top >= 0) {
            return stack[top--];
        }
        return '\0';
    }
    public char peek() {
        if (top >= 0) {
            return stack[top];
        }
        return '\0';
    }
    public boolean isEmpty() {
        return top == -1;
    }
}
public class yash {
    public static int precedence(char c) {
        if (c == '+' || c == '-') {
            return 1;
```



```
        }
        if (c == '*' || c == '/') {
            return 2;
        }
        return 0;
    }
    public static String infixToPostfix(String infix) {
        Stack stack = new Stack(infix.length());
        StringBuilder postfix = new StringBuilder();
        for (int i = 0; i < infix.length(); i++) {
            char c = infix.charAt(i);
            if (Character.isLetterOrDigit(c)) {
                postfix.append(c);
            } else if (c == '(') {
                stack.push(c);
            } else if (c == ')') {
                while (!stack.isEmpty() && stack.peek() != '(') {
                    postfix.append(stack.pop());
                }
                stack.pop();
            } else {
                while (!stack.isEmpty() && precedence(c) <= precedence(stack.peek())) {
                    postfix.append(stack.pop());
                }
                stack.push(c);
            }
        }
        while (!stack.isEmpty()) {
            postfix.append(stack.pop());
        }
        return postfix.toString();
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter infix expression:");
        String infix = sc.nextLine();
        String postfix = infixToPostfix(infix);
        System.out.println("Postfix expression: " + postfix);
    }
}
```

Output -



```
Listening on 52315
User program running
Enter infix expression:
A+B*(C-D)

User program finished
Postfix expression: ABCD-*+
```