Infix to Postfix conversion

Infix Expression:

Infix expressions are those expressions in which the operator is written in-between the two or more operands. Usually, we use infix expression. For example, consider the following expression.

- a+b
- a/2+c*d-e*(f*g)
- a*(b+c)/d

Postfix Expression

Postfix expressions are those expressions in which the operator is written after their operands. For example, consider the following expression.

- ab+
- abc/de+*+f-
- ab+cd-*

Problem description: A program in java to convert an infix expression to postfix expression.

Output: The postfix form of the infix expression entered by user.

Program in java:

```
import java.io.*;
class Stack
{
    char[] a=new char[100];
    int top=-1;
    void push(char c)
    {
        try
        {
            a[++top]= c;
        }
        catch(StringIndexOutOfBoundsException e)
        {
            System.out.println("Stack size cannot be more than 100 elements");
            System.exit(0);
        }
    }
    char pop()
    {
```

```
        return a[top--];
    }
    boolean isEmpty()
    {
        return (top == -1);
    }
    char peek()
    {
        return a[top];
    }
}
public class InfixToPostfix
{
    static Stack operators = new Stack();
    public static void main(String args[]) throws IOException
    {
        String infix;
        InputStreamReader intr = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader (intr);
        System.out.println("Enter the infix expression you want to convert: ");
        infix = br.readLine();
        System.out.println("Postfix expression for the given infix expression is:"
        + toPostfix(infix));
    }
    public static String toPostfix(String infix)
    //converts an infix expression to postfix
    {
        char symbol;
        String postfix = "";
        for(int i=0;i<infix.length();++i)
        {
            symbol = infix.charAt(i);
            //if it's an operand, add it to the string
            if (Character.isLetter(symbol))
                postfix = postfix + symbol;
            else if (symbol=='(')
            //push (
            {
                operators.push(symbol);
            }
            else if (symbol==')')
            //push everything back to (
            {
                while (operators.peek() != '(')
                {
                    postfix = postfix + operators.pop();
                }
                operators.pop();
            }
            else
            {
                while (!operators.isEmpty() && !(operators.peek()=='(') &&
                 precedence(symbol) <= precedence(operators.peek()))
                    postfix = postfix + operators.pop();
                operators.push(symbol);
            }
        }
        while (!operators.isEmpty())
            postfix = postfix + operators.pop();
```

```
        return postfix;
    }
    public static int precedence(char x)
    {
        if (x == '+' || x == '-')
            return 1;
        if (x == '*' || x == '/' || x == '%')
            return 2;
        return 0;
    }
}
```

Output:

1.

```
"C:\Program Files\Java\jdk-11.0.3\bin\java.exe" "-javaagent:C:\Program Files
Enter the infix expression you want to convert:
a/(b+c)*d-e*f
Postfix expression for the given infix expression is:abc+/d*ef*-

Process finished with exit code 0
```

2.

```
"C:\Program Files\Java\jdk-11.0.3\bin\java.exe" "-javaagent:C:\Program Files\
Enter the infix expression you want to convert:
(p-q)*r+s/t
Postfix expression for the given infix expression is:pq-r*st/+

Process finished with exit code 0
```