Lab Exercise 14 Student Name: ChenYu Yang Student ID: 1670203

Download the source file, EvaluateExpression.javaPreview the document, from the "demo files" folder under Files. Run the source file from your IDE.

Review the source file, discuss with your peer and answer the following questions:

1. Why create two stacks to store operands and operators separately? Is it fine to use one stack instead?

***Answer:*** No, it can not use one stack instead. The order in which the operators appear need same as the order of the actual arithmetic operations, it needs to have the operators with higher precedence and parentheses appear first. So we need to use one stack to hold operators that have not been sent to the suffix expression, it called the operator stack.

And in the calculation process, we also need to use a stack to retain the operands in the suffix expression that have not been involved in the operation, it called the operand stack.

So we have to use two stacks to store operands and operators separately.

1. Before push a '+' or '-' operator into the operator stack, why it should pop and process all operators (+, -, \*, /) by invoking,

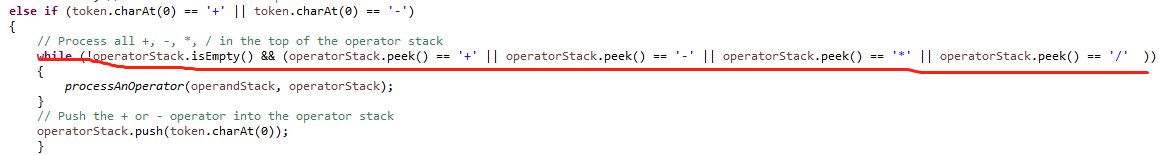
processAnOperator(operandStack, operatorStack);

***Answer:*** Because in the calculation process, you also need to use a stack to retain the operands in the suffix expression that have not been involved in the operation. We need to check whether all operators in the suffix expression, and "\*, /, % " is in the top of the operator stack. So we should pop and process all operators (+, -, \*, /) by invoking.

1. In Assignment 14, you need to add two more operators, '%' (mod) and '^' (exponent). How will you work on it?

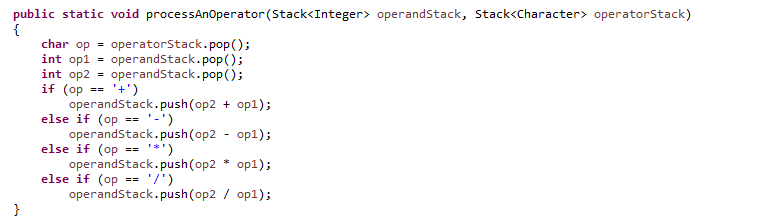
***Answer:***

1.Before the setence “processAnOperator(operandStack, operatorStack);” we need add “operatorStack.peek() == '^' || operatorStack.peek() == '%'” in the red pen.



2.In the **public** **static** **void** processAnOperator(Stack<Integer> operandStack, Stack<Character> operatorStack)

We need add “ else if (op == '^') operandStack.push((int)Math.pow(op2, op1));” and “else if (op == '%') operandStack.push(op2 % op1);” in the end.



3.In the **public** **static** String insertBlanks(String s)

We need add “ s.charAt(i) == '^' || s.charAt(i) == '%'” in the condition of if statement.

