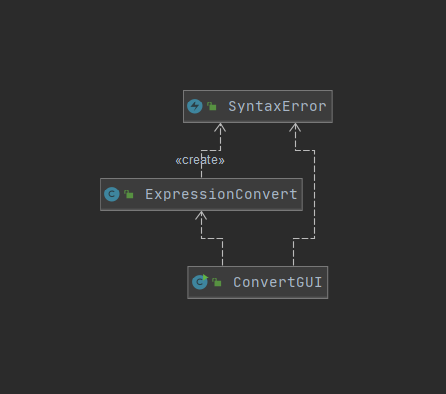
# UML Class Diagram:

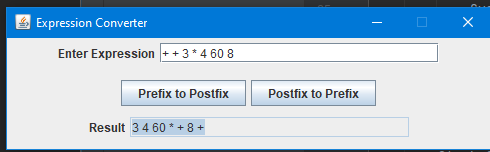
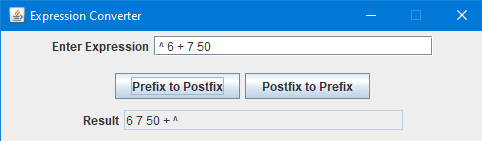
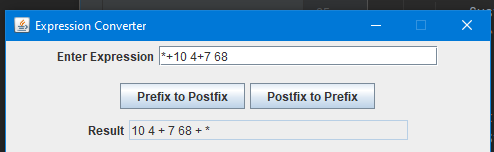
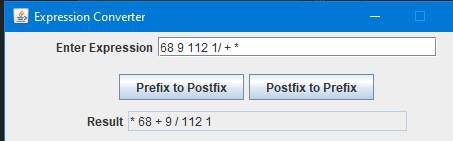


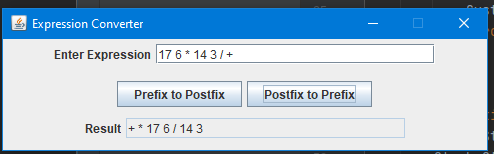
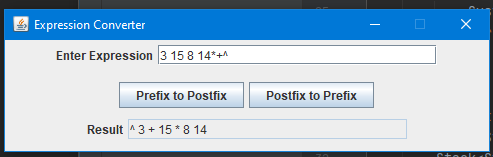
# Test Cases:

## Test Case 1:

Description: 6 different expressions, 3 prefix and 3 postfix. 1 of each prefix and postfix containing: spaces between everything, no spaces (except between integers), and a mix of spacing for the last one.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Converting Prefix to Postfix | | | | |
| Test | input | Expected Output | Actual Output | Pass/Fail |
| Prefix A: | + + 3 \* 4 60 8 | 3 4 60 \* + 8 + | 3 4 60 \* + 8 + | Pass |
| Prefix B: | \*+10 4+7 68 | 10 4 + 7 68 + \* | 10 4 + 7 68 + \* | Pass |
| Prefix C: | ^ 6 + 7 50 | 6 7 50 + ^ | 6 7 50 + ^ | Pass |
| Converting Postfix to Prefix | | | | |
| Test | input | Expected Output | Actual Output | Pass/Fail |
| Postfix A: | 17 6 \* 14 3 / + | + \* 17 6 / 14 3 | + \* 17 6 / 14 3 | Pass |
| Postfix B: | 3 15 8 14\*+^ | ^ 3 + 15 \* 8 14 | ^ 3 + 15 \* 8 14 | Pass |
| Postfix C: | 68 9 112 1/ + \* | \* 68 + 9 / 112 1 | \* 68 + 9 / 112 1 | Pass |

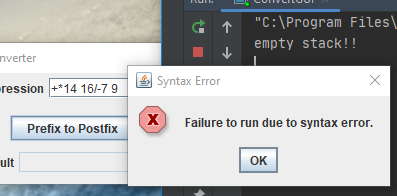
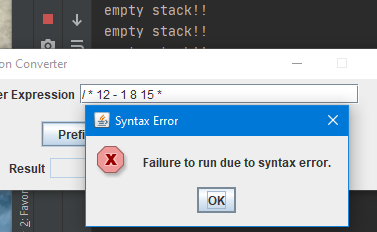
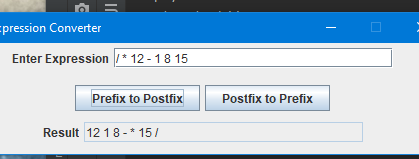
Test case 1 Images:

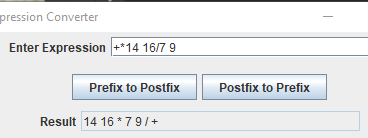


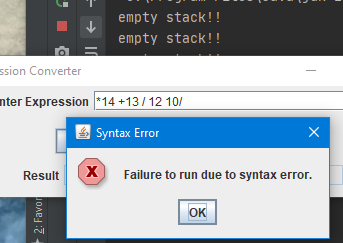
## Test Case 2:

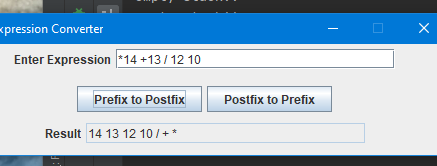
Description: 6 different expressions, 3 prefix and 3 postfix. Testing if different expressions with too many operators cause syntax error to display. Also, I ran a proper conversion after to make sure program still functions after error corrected (without closing window and re-running program).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Converting Prefix to Postfix | | | | |
| Test | input | Expected Output | Actual Output | Pass/Fail |
| Prefix A: | +\*14 16/7 9 | Syntax Error(popup)  Empty stack print in terminal | Syntax Error(popup)  Empty stack print in terminal |  |
| Prefix B: | / \* 12 - 1 8 15 \* | Syntax Error(popup)  Empty stack print in terminal | Syntax Error(popup)  Empty stack print in terminal |  |
| Prefix C: | \*14 +13 / 12 10/ | Syntax Error(popup)  Empty stack print in terminal | Syntax Error(popup)  Empty stack print in terminal |  |
| Converting Postfix to Prefix | | | | |
| Test | input | Expected Output | Actual Output | Pass/Fail |
| Postfix A: | 1 5 7 \*- + 8 + | Syntax Error(popup)  Empty stack print in terminal | Syntax Error(popup)  Empty stack print in terminal |  |
| Postfix B: | 3 5 8 4\*+^- | Syntax Error(popup)  Empty stack print in terminal | Syntax Error(popup)  Empty stack print in terminal |  |
| Postfix C: | 9 4 \*+ 3 2 \* + | Syntax Error(popup)  Empty stack print in terminal | Syntax Error(popup)  Empty stack print in terminal |  |

Test case 2 images, in order(left to right):

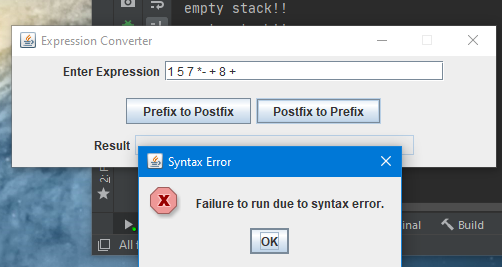
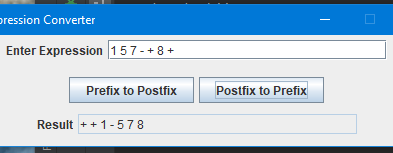
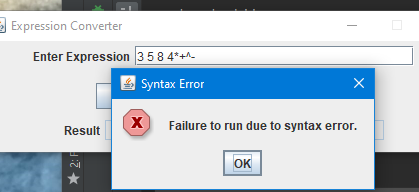
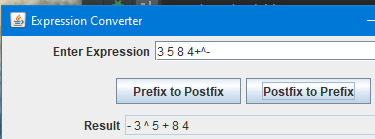


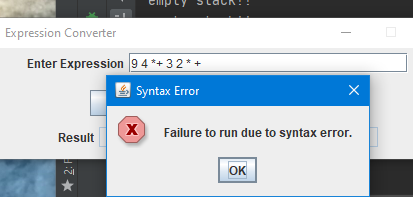
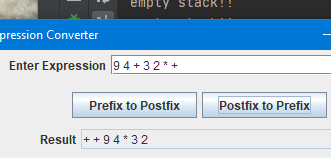










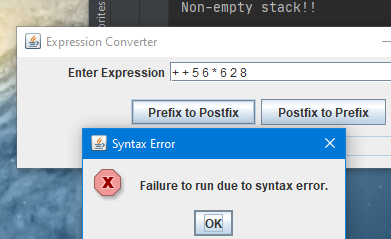
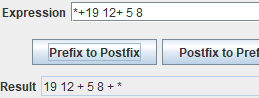
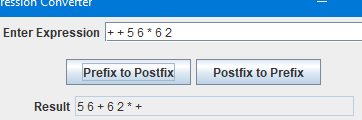


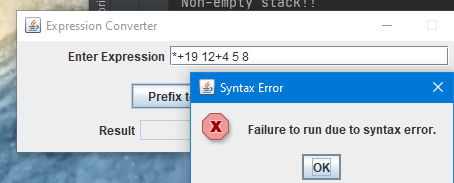


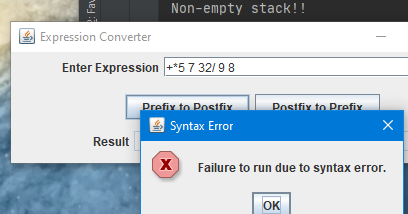
## Test Case 3:

Description: 6 different expressions, 3 prefix and 3 postfix, a non-empty stack will occur if there are extra operands. Will mix up spacing on each run too.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Converting Prefix to Postfix | | | | |
| Test | input | Expected Output | Actual Output | Pass/Fail |
| Prefix A: | + + 5 6 \* 6 2 8 | Syntax error popup  Non-empty stack print in terminal | Syntax error popup  Non-empty stack print in terminal |  |
| Prefix B: | \*+19 12+4 5 8 | Syntax error popup  Non-empty stack print in terminal | Syntax error popup  Non-empty stack print in terminal |  |
| Prefix C: | +\*5 7 32/ 9 8 | Syntax error popup  Non-empty stack print in terminal | Syntax error popup  Non-empty stack print in terminal |  |
| Converting Postfix to Prefix | | | | |
| Test | input | Expected Output | Actual Output | Pass/Fail |
| Postfix A: | 12 5 10 - 4 + 5 + 5 | Syntax error popup  Non-empty stack print in terminal | Syntax error popup  Non-empty stack print in terminal |  |
| Postfix B: | 5 6^5+5 | Syntax error popup  Non-empty stack print in terminal | Syntax error popup  Non-empty stack print in terminal |  |
| Postfix C: | 2 8 -7 9+ ∗ | Syntax error popup  Non-empty stack print in terminal | Syntax error popup  Non-empty stack print in terminal |  |

Test case 3 images in order left to right:





# 

# 

# 

*Postfix to Prefix*

# Lessons Learned:

My biggest lesson learned from this project is to make sure I test every case before I think all the coding for a project is done and wait till the due date to run all test cases and put word document together. I thought I tested everything that could happen yesterday, but completely missed the fact that I was only testing with single digit integers. My code fell apart today when I tested with numbers like 10 and 15. I was able to reuse some of my previous code, but had to rethink how I would take an input and separate it into operands and operators. I found that using a scanner was not going to work for me, so I decided just to loop through the input. This turned out to work just fine, thankfully.