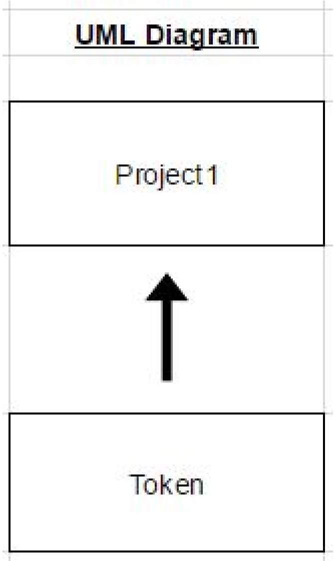
Project 1

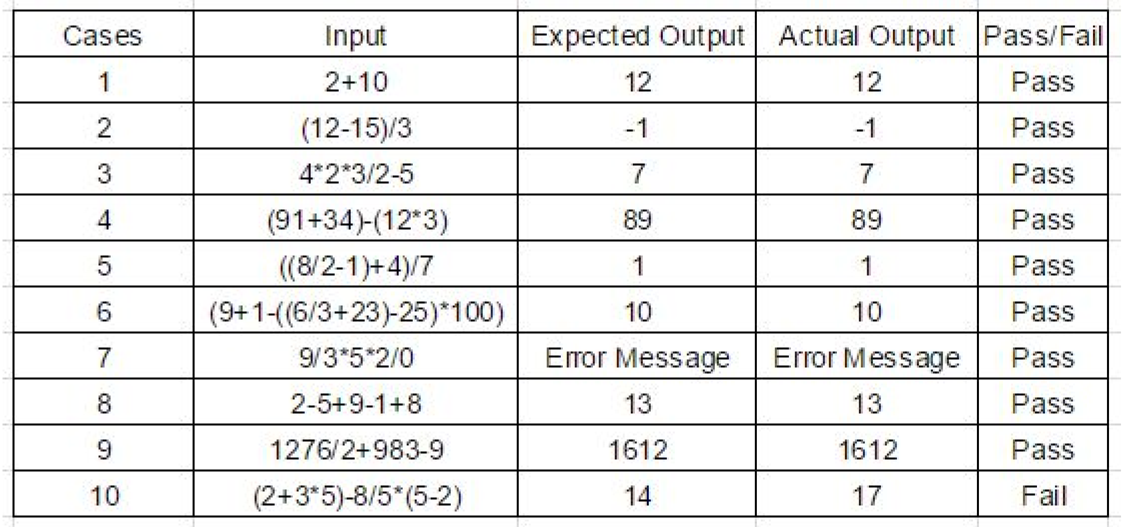
1.



(Not sure if this is correct, as it seems quite simple - fingers crossed!)

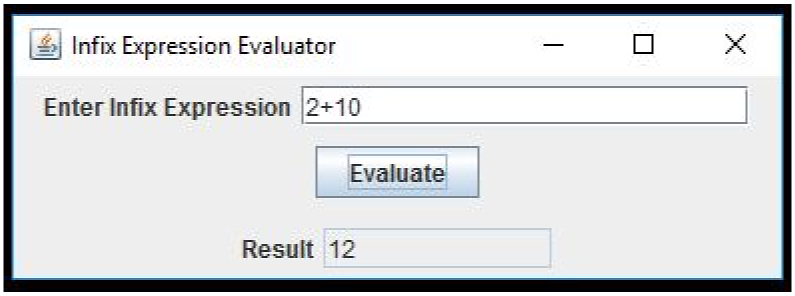
2.

Test Table



Cases	Input	Expected Output	Actual Output	Pass/Fail
1	2+10	12	12	Pass
2	(12-15)/3	-1	-1	Pass
3	4*2*3/2-5	7	7	Pass
4	(91+34)-(12*3)	89	89	Pass
5	((8/2-1)+4)/7	1	1	Pass
6	(9+1-((6/3+23)-25)*100)	10	10	Pass
7	9/3*5*2/0	Error Message	Error Message	Pass
8	2-5+9-1+8	13	13	Pass
9	1276/2+983-9	1612	1612	Pass
10	(2+3*5)-8/5*(5-2)	14	17	Fail

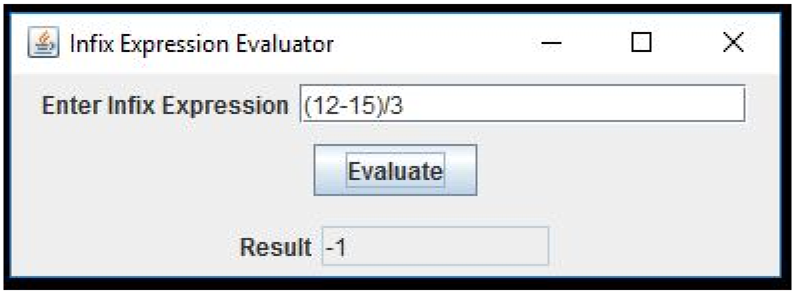
Case 1 - Pass : Simple addition



Enter Infix Expression: 2+10

Result: 12

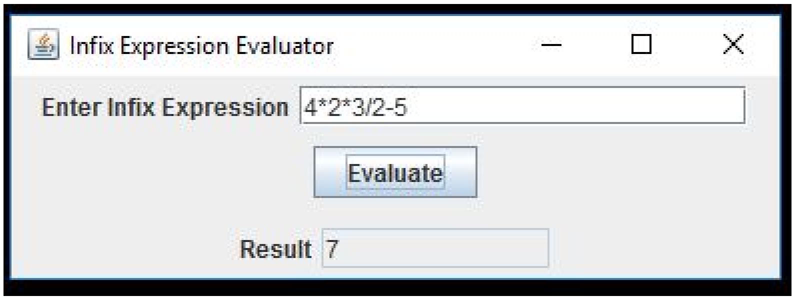
Case 2 - Pass : Mixing parentheses with non parenthetical inputs



Enter Infix Expression: (12-15)/3

Result: -1

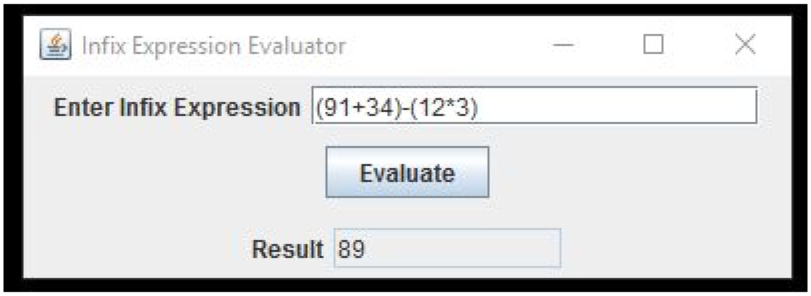
Case 3 - Pass : Both multiplication and division with subtraction as well



Enter Infix Expression: 4*2*3/2-5

Result: 7

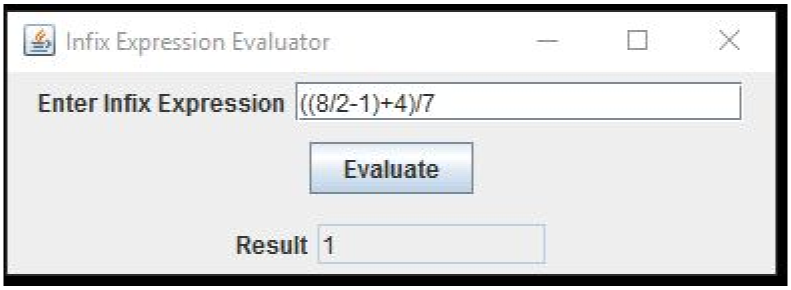
Case 4 - Pass : Usage of two parenthesized inputs with subtraction



Enter Infix Expression: (91+34)-(12*3)

Result: 89

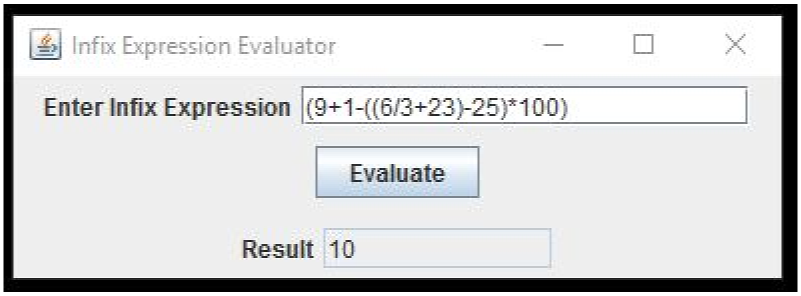
Case 5 - Pass : Usage of parenthesis within parenthesis as well as input outside of parenthesis



Enter Infix Expression: ((8/2-1)+4)/7

Result: 1

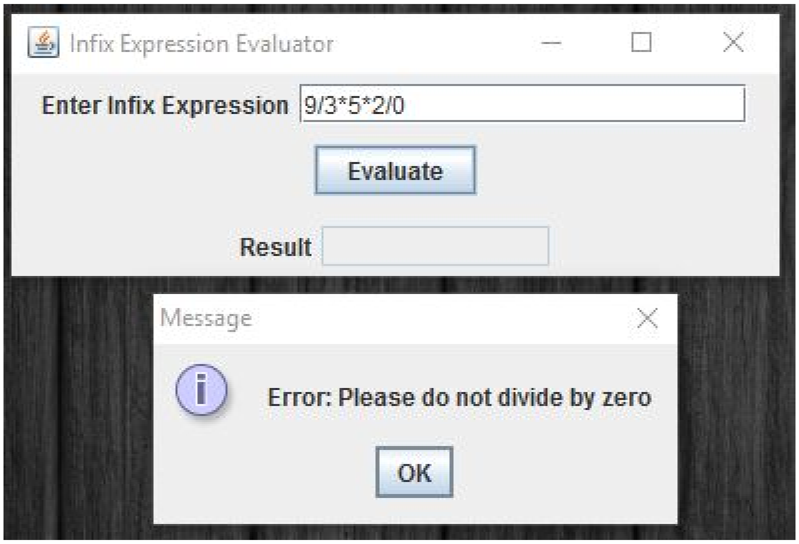
Case 6 - Pass : Similar to case 5 but with larger and more inputs



Enter Infix Expression (9+1-((6/3+23)-25)*100)

Result 10

Case 7 - Pass : Proof that the divide by zero error message is working

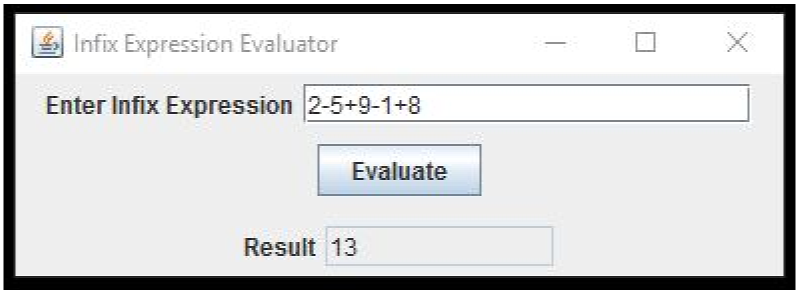


Enter Infix Expression 9/3*5*2/0

Result

Error: Please do not divide by zero

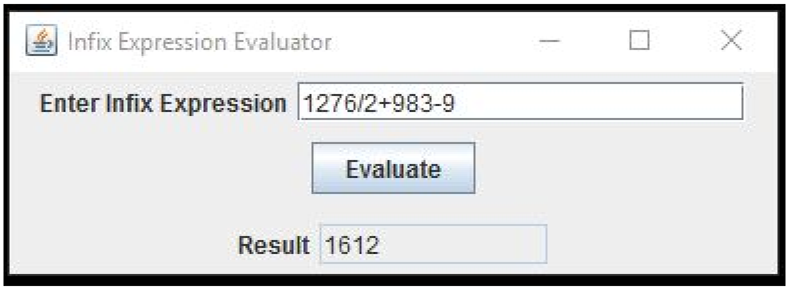
Case 8 - Pass : Mixture of addition and subtraction, mainly testing precedence as this was a hurdle for me. At one point in my testing this type of expression would end up getting working from right to left which was problematic



Enter Infix Expression 2-5+9-1+8

Result 13

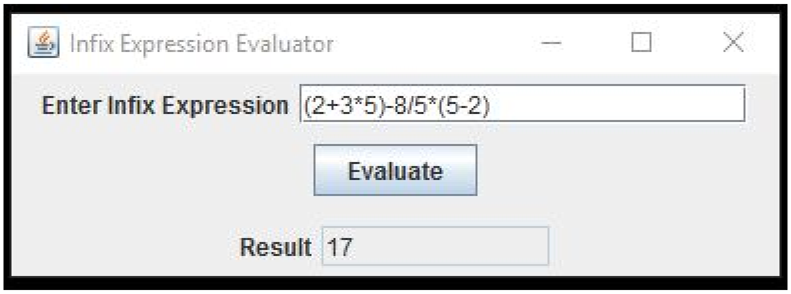
Case 9 - Pass : Larger numbers while mixing three kinds of operators



Enter Infix Expression 1276/2+983-9

Result 1612

Case 10 - Fail : I found it funny that this one ended up failing, as before I used the setPrecedence method that you provided me, this expression was giving me the correct answer of 14. I did notice that it seemed to be not be taking division as an operator with equal precedence to multiplication so when calculating ‘5*(3)’ is calculated ending in an ‘8/15’ value of integer 0. With more testing I may have been able to fix this but it is nearly midnight!



Enter Infix Expression (2+3*5)-8/5*(5-2)

Result 17

3.
I feel like I learned a great deal about how stacks work during this past couple of weeks working on this project. I relearned how to make a GUI as the last time I made one was in the latter half of the Fall semester. That came back pretty quickly and I had it done in a couple of days. I tried to do it as simply as possible in a way that emulated the look of the example that was provided. Most of my learning happened while implementing the given pseudo code. I learned how to tokenize a string with the StringTokenizer library imported as well as how to insert that into a stack. I started out with a good deal of calculation code that was repeated quite a bit so I was able to reduce my code by creating a method called ‘compute’ which I replaced the repetitious parts of my code with. I also learned a good way of using a method to set precedence of operators, although this is one part that I still think is flawed, as I still am getting on failed test case but have run out of time and need to get this turned in! Probably the most useful tool I learned to use during this project was the debugger. This tool is amazing and it really helps to know which value is attached to which variable step by step going through the code.