Expression Tree Problem

Design Consideration

Expression Tree		Tree Node
- root_hode : <holds the head node or root node of expression tree>		- data : <holds value either operator or operand> - left_ref: <holds left reference> - right_ref : <holds right reference>
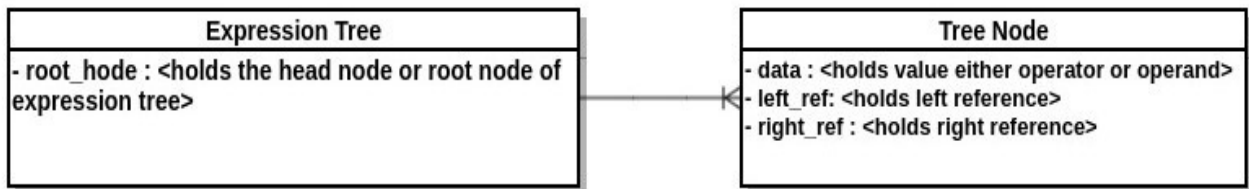
In this assignment tree data structure is used to represent the expression

It has basically 2 operations

1) Evaluate Expression

- It first converts an infix string to a postfix string
- and uses postfix string to generate an expression tree
**- and lastly it evaluates this expression tree to get the final result of the
  expression**

2) Display Tree

It pretty prints the expression tree level wise

**Note : For testing please only enter the operators from ['+', '-', '/', '*', '(', ')']
with proper paranthesis**

e.g   Invalid : 1+2*3/11
     Valid: (1 + 2) * (3 / 11)

- Run Instructions
I have created two scripts run and test for ease of use and quick testing
step 1 : scripts/run – this will run the app and show user dialog to enter input

step 2: Enter your options accordingly

step 3: scripts/test – this will run all the unit testcases

Note: If script is not executing just need to make it executable by

" chmod +x scripts/run "

Output Screens

```
 Please enter the expression string, pls enter operator from only ['+', '-', '/', '*', '(', ')'] with proper parenthesis : ((15 / (7 - (1 + 1) ) ) * 3 )
- (2 + (1 + 1))

 The Expression Tree is created successfully!


1) Evaluate Expression

2) Display Expression Tree

3) Exit

Please Enter your choice : 2

 Choice entered : Display Expression Tree

                                    /------------------------      -      ------------------------\
                                    *                                                             +
                        /-----------      -----------\                         /-----------      -----------\
                        /                             3                         2                 +
                /------      ------\                                                       /------      ------\
                15                -                                                        1                 1
                        /--- ---\
                        7      +
                            /- -\
                            1   1
```