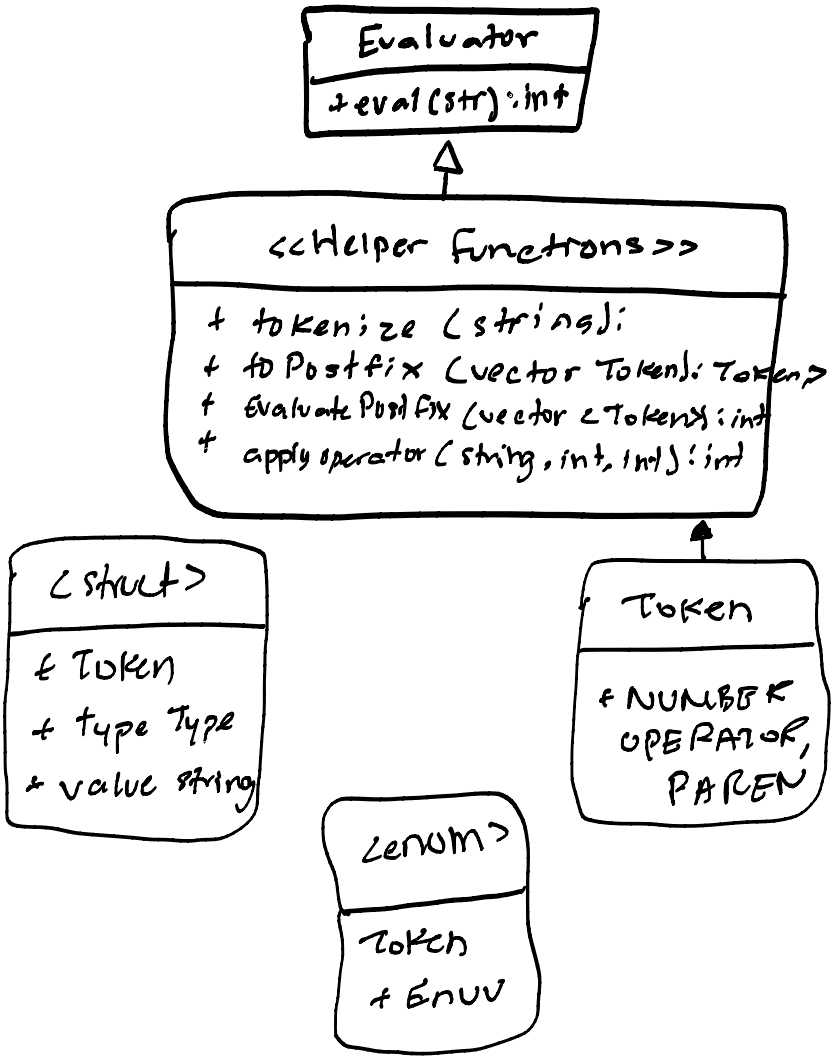
Oh May Ishar
Project 1A
CS 303- Data Structure

A UML class diagram:



This diagram shows how the program is built. The main class is Evaluator, and it has one public function called eval() that runs everything. Inside the code, there are helper functions that do things like break the expression into pieces (tokenize), change it to postfix form (toPostfix), and solve it using a stack (evaluatePostfix). There's also a Token struct to hold each part of the expression, and TokenType is used to tell if it's a number, operator, or parenthesis.

Efficiency of Algorithms:

The main functions in this program all run in linear time, based on how much tokens are in the expression:

- **tokenize( ) → O(n)**
    - Goes through the string once and creates tokens.
- **toPostfix( ) → O(n)**
    - Uses a stack to convert infix to postfix by checking precedence.
- **evaluatePostfix( ) → O(n)**
    - Evaluated the postfix with a stack, one pass through.
- **eval( ) → O(n)**
    - Just calls the other three in order

I don't think you can make this any faster, since each part only loops through the expression once. I believe it is already efficient for this type of task.

Contribution:

This project was worked on by myself, I didn't have any teammates. I wrote the code, tested it, added comments, and made sure it ran. I also created the report and created the README file for github, as well as writing this report.

Reference:

- **Class lectures and slides on expression evaluation and stacks**
    - Used to understand how to use stacks to convert infix to postfix and evaluate expressions

- **[Infix to postfix conversion]**

  https://www.geeksforgeeks.org/convert-infix-expression-to-postfix-expression/
    - Helped me understand how operator precedence and the shunting yard algorithm work.

- **[Stack data structure] https://www.programiz.com/dsa/stack**
    - Used to review how stacks work and how to implement them in c++.

- **[UML class Diagram tutorial] https://www.lucidchart.com/pages/uml-class-diagram**
    - Helped me understand how to design and format a UML class diagram