CS 340 Spring 2022
Programming Quiz 3

Due 11:59 Tuesday March 1
Expression Tree

- Implement and test the ExpressionTree class shown on the following slides.
- The constructor can assume the expression passed as a parameter is syntactically correct.
- Tokens (operands and operators) are separated by blanks.
- evaluate returns the integer value of the expression tree.
- All calculations will be done with integer arithmetic.
Expression operators

- The operators in precedence order are
  - ! (unary minus)
  - ^ (exponentiation) (you can assume exponents are greater than or equal to 0.
  - *, /, %
  - +,-

- Exponentiation and unary minus are right associative
- All other operators are left associative
- Expressions can use parentheses to change the order of expression evaluation
Expression operands

- The operands will be ints represented by strings of digits.
public class ExpressionTree {

    private class Node {
        private Node left;
        private String data;
        private Node right;

        private Node(Node l, String d, Node r) {
            left = l;
            data = d;
            right = r;
        }
    }

    private Node root;
}
public ExpressionTree(String exp) {
    //PRE: exp is a legal infix expression with tokens separated by blanks
    //Build an expression tree from the expression exp
}
public int evaluate() {
    // return the int value of the expression tree
    return evaluate(root);
}

private int evaluate(Node r) {
    // return the int value of the expression tree with root r
    
}
public String toPostfix() {
    //return the postfix representation of the expression tree
    return toPostfix(root);
}

private String toPostfix(Node r) {
    //return the postfix representation of the tree with root r
}
public static void main(String args[]) throws IOException {
    //used to test expression tree
}

Example Expressions

3 ^ 2 ^ 3
100 ÷ !20 + 3 * 73
2 * (2 + (3*(4-2))) ^ (10 - 8)
!!30
Programming Quiz 3 Submission

- Upload one zip file to Canvas. The zip file must contain only one file called `ExpressionTree.java`. Do not upload your whole Eclipse project!

- Include a comment with your name at the top of the file and add comments for any private instance variables or method that you add.

- You can use main to test your code. I will use my own test driver.