CSC3100 Assignment 1

Important Notes:

1. The assignment is an individual project, to be finished on one's own effort.
2. Submissions before the deadline 6pm Oct. 18, 2021 are treated as normal submissions. Submissions within one week after the deadline are treated as late submissions.
3. Plagiarism is strictly forbidden, regardless of the role in the process. Notably, ten consecutive lines of identical codes are treated as plagiarism.

Marking Criterion:

1. The full score of the assignment is 100 marks.
2. Normal submission: There are two test cases (test A and test B, thereafter). A submission passes a test if and only if all 100 expressions in each test are correctly evaluated. 80 marks are given if the submission passes test A. 100 marks are given if the submission passes both test A and test B.
3. Resubmission: For normal submissions that fail in test A, a resubmission is allowed within one week after the marking. 60 marks are given if the resubmission passes test A. 75 marks are given if the resubmission passes both test A and test B.
4. Late submission: 60 marks are given if the submission passes test A. 75 marks are given if the submission passes both test A and test B. No resubmission is allowed.
5. Zero mark is given if: there is no (normal or late) submission; a normal submission fails test A and the resubmission fails test A; a normal submission fails test A and there is no resubmission; a late submission fails test A.
6. In case of identical copies, both submissions will be marked as zero. Similar rules apply to other cases of plagiarism found.

Running Environment:

1. The submissions will be evaluated in Java environment under Linux platform.
2. The submission is acceptable, if it runs in any of recent versions of Java SDK environment. These versions include from Java SE 8 to the most recent Java SE 17.
3. The submission is only allowed to import four packages of (java.lang.*; java.util.*; java.math.*; java.io.*) included in Java SDK. No other packages are allowed.
4. In each test, the program is required to evaluate 20 expressions within 1 second of time, with no more than 128MB memory.
5. Each student will have an opportunity to test his/her program in the evaluation platform before the submission. Detailed instructions of testing will be given around Oct. 8.

Submission Guidelines:

1. Detailed submission guideline will be given in a separate manual around Oct. 8.
2. Inconsistency with or violation from the guideline leads to marks deduction.

Functional Requirement:

1. The program evaluates the value of math expressions, and outputs an integer value or "invalid".
2. In test A, each math expression includes (see the example below):
   2.1 positive integers;
   2.2 (no more than five) operators of "+" (addition), "-" (subtraction), "*" (multiplication) and "/" (division).
3. In test A, all expressions are valid. The output is an integer value after rounding.

An example of input file for test A	Expected output file:
1+2	3
3+4/3+7	11
1*2*3*4+5	29
2*3+4*5+6/7	27
1/2-1*3/3	0
1/2-1*4/3	-1
…	…

4. In test B, each math expression includes (see the example below):
   4.1 numbers (integers and doubles);
   4.2 (no more than ten) operators of "+" (addition), "-" (subtraction), "*" (multiplication) and "/" (division);
   4.3 (no more than three) functions including "sin" (sine function), "cos" (cosine function), "tan" (tangent function) and "sqrt" (square root function).
   4.4 "(" and ")" (brackets);
   4.5 possibly blank space.
5. In test B, the expression may not be valid. Correspondingly, the output is either the integer value after rounding if the expression is valid, or "invalid" if the expression is not valid.

An example of input file for test B	Expected output file:
1+2.0*sin(37+(25*3))	-1
(2+ 3.50)*4*sqrt(sin(1.5))	22
-3+4/ (2.5+3.7)	-2
(-3+4)/2.5+3.9	4
1.2-3.5*5.2-13.2	-30
1.2-3.5*5.2-13.7	-31
2.3*5*7 - 12*9/8	67
-sin(3.5-sqrt(4)) + cos(tan(2.5))	0
sqrt(-1)	invalid
2.5 4.0	invalid
…	…

Program Template:

Each submission is expected to strictly follow the template to implement the required function.

```java
import java.io.*;

public class TestMathExpr {
    public static void main(String[] args) throws Exception {
        FileReader freader = new FileReader(args[0]);
        BufferedReader breader = new BufferedReader(freader);
        FileWriter fwriter = new FileWriter(args[1], false);
        BufferedWriter bwriter = new BufferedWriter(fwriter);
        String str = breader.readLine();
        while (true) {
            double result = MathExpr.parse(str);
            if (!Double.isNaN(result)) {
                bwriter.write(String.valueOf(Math.round(result)));
            } else {
                bwriter.write("invalid");
            }
            str = breader.readLine();
            if (str == null) break;
            bwriter.newLine();
        }
        freader.close();
        breader.close();
        bwriter.flush();
        bwriter.close();
    }
}
```

```java
public class MathExpr {
    public static double parse(String str) {
        // implement the code here
        // evaluate a math expression, and return the value
        // if the expression is invalid, return NaN.



        // to be revised
        return 0.0d;
    }
}
```

The main function is in *TestMathExpr.java*. In line 11 of *TestMathExpr.java*, an expression string is evaluated, which returns the value of the expression. Each submission is required to:

1. Modify the *MathExpr.java* and implement the *parse()* function;
2. If needed, new classes and functions can be added;
3. No modification of *TestMathExpr.java* is allowed.

The command used to run the program is: *java TestMathExpr input.txt output.txt*, where *input.txt* is the input file of expressions, and *output.txt* is the output file of values.

Appendix

1. TestMathExpr.java
2. MathExpr.java
3. inputA.txt
4. outputA.txt
5. inputB.txt
6. outputB.txt