CSC3100 Assignment 1

Important Notes:

1. The assignment is an individual project, to be finished with one's own effort.
2. The submission deadline is 6 PM Feb. 20, 2023 (Monday), Beijing Time. This is a firm deadline. No late submissions are accepted.
3. Please also submit your final program and rename it as "StudentID_A1.java" on the blackboard. For example, a student whose Student ID is "120000001" should submit the program named as "120000001_A1.java". **You don't need to consider the consistency of class name and file name** since we won't run the code you submitted on the blackboard. File misnaming or no submission on the blackboard will lead to **5 demerit points.**
4. OJ website: http://oj.cuhk.edu.cn/. If you are off campus, please use VPN to access the OJ.
5. Access code: csc3100as1
6. Each student is only permitted to submit code to OJ **up to FIVE times**. Only the **last submission** will be used in evaluation of assignment marks.
7. Plagiarism is strictly forbidden, regardless of the role in the process. Notably, ten consecutive lines of identical codes are treated as plagiarism. Depending on the seriousness of the plagiarism, 30%-100% of marks will be deducted.

Marking Criterion:

1. The full score of the assignment is 100 marks.
2. There are two test cases (test A and test B, thereafter). Each test has 50 marks.
3. Zero mark is given if: there is no submission; a normal submission fails both test A and test B.
4. We will use the "pretest + system test" strategy to grade your programs. Before the deadline, OJ will only run the pretest. **The score you get in the pretest doesn't represent your final score.** After the deadline, we will run another set of test cases for the final grading.

Running Environment:

1. The submissions will be evaluated in the course's OJ system running Java SE version 17 and Linux platform.
2. 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.
3. If you have any questions about OJ, please refer the wiki (http://oj.cuhk.edu.cn/wiki#/) first. You can also ask for help via Wechat group or email.

Submission Guidelines:

1. Inconsistency with or violation from the guideline leads to marks deduction.
2. It is the students' responsibility to read this assignment document and submission guidelines carefully and in detail. No argument will be accepted on issues that have been specified clearly in these documents.

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 numbers (integers and doubles);
   2.2 (no more than five) operators of "+" (addition), "-" (subtraction), "*" (multiplication) and "/" (division);
   2.3 possibly blank space.
3. In test A, all expressions are valid. The output is an integer value after rounding.

Sample Input:	Expected Output:
1+2.0	3
-3+4/ 2.5+3.7	2
-3+4/2.5+3.9	3
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

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 ten) 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. So the output is either the integer value after rounding if the expression is valid, or "invalid" if the expression is not valid.

Clarifications:

1. Each math expression in the test cases must contain at least one number. In other words, expressions like "()" and " " will not be tested. An expression like " 3.0 " is valid, so you should output "3";
2. Operators ("+", "-", "*", "/") in a valid expression will not appear continuously. So, expression like "1*-3" is considered invalid in our test. The valid expression is "1*(-3)", which you should output "-3";
3. There will be no additional sign "+" before the positive number in the test cases;
4. If you get confused about the validity of the math expression, you can test it in python/MATLAB.

Sample Input:	Expected Output:
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:

A skeleton of the program is given below:

```java
import java.util.*;

public class TestMathExpr {

    public static double parse(String str) {
        // evaluate a math expression, and return the value
        return 0.0d;
    }

    public static void main(String[] args) throws Exception {
        Scanner input = new Scanner(System.in);
        while (input.hasNextLine()) {
            double result = parse(input.nextLine());
            System.out.println(String.valueOf(Math.round(result)));
        }
    }
}
```

In line 13 of TestMathExpr.java, an expression string is evaluated, which returns the value of the expression.

The above template is for reference only, you do not need to strictly follow the template format. This template can be used directly for test A. For the invalid cases of test B, you can modify the template to solve these cases.