

CS 570: Final Project

(In lieu of the Final exam—30%)

Due: By May 14, 2022.

(Late submissions won't be accepted)

1 Assignment Policies

Collaboration Policy. The project is to be completed individually: each student must hand in the student's own answers. It is acceptable for students to collaborate in understanding the material but not in solving the problems or programming. Use of the Internet is allowed, but that should not include searching for existing solutions.

Under absolutely no circumstances code can be exchanged among students. The code presented in the book and in class can be used.

2 Assignment

You have to implement a calculator that would parse and evaluate an arbitrary expression that may

- 1) Rational numbers (e.g., 4.5669 or -3.5);
- 2) Parentheses '(' and ')';
- 3) Operators "+", "-", "*", "/", and "^". The latter operator is that of exponentiation, which, has the highest precedence. **Your code must handle rational exponents—including the negative ones.**

Your program **must**¹ use a recursive-descent technique, as explained in class, based on the following grammar for expressions, as discussed in class:

```
<expression> → <term>
<expression> → <term> '+' <term>
<expression> → <term> '-' <term>
<term> → <factor>
<term> → <factor> '*' <factor>
<term> → <factor> '/' <factor>
<factor> → <number>
<factor> → '(' <expression> ')'
<number> → <number> '.' <number>
<number> → <digit> <number>
<digit> → '0', '1', '2' ... '9'
```

Your program must correctly handle all error conditions (it must never crash!). You may use all Java mathematical libraries available to you.²

3 Additional features

If your program is ready before the deadline, you can get extra **15 points** by allowing your expressions to contain identifiers. In other words, your program should recognize an *assignment*: **<identifier> '=' <expression>** and process it by printing the value of the identifier. For example, you could write `x = 20`, which will result in your program displaying 20. Once an identifier has been assigned a value, it can be used in subsequent expressions (e.g., `x+3`). A subsequent assignment can change the value of the identifier.

You should store all identifiers in a hash table. An identifier can be created or modified only via the assignment. If a non-existing identifier is referenced, your program must return an error.

¹ If the code does **not** implement this technique (for example, if it uses the direct stack operations as shown in the textbook), the grade for this assignment will be 0.

² For instance to implement rational exponentiation $a^b = e^{b \ln a}$ you may use the Java Math methods for finding natural logarithms and exponentiation..

Extra points for this feature will be given **only** if the rest of the program is working.

4 Submission instructions

Submit a single file named `Expression.zip` through Canvas that includes *Calculator.java* and *CalculatorTest.java* with your test cases..

Make sure that you test your code well! Up to 20 points will be deducted for each of the following problems:

- 1) An ill-formed expression is not detected or a wrong error message is given;
- 2) The calculation result is wrong.

The code must be well-commented.

The code that does not compile or crashes will be found unacceptable.