CS 570: Midterm Project

(In lieu of the Midterm Exam, but with 60% weight)

Due: March 25, 11:59pm

1   Assignment Policies

Collaboration Policy. Homework 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 evaluate an arbitrary long expression that contains

1) Rational numbers (e.g., 4.5669);
2) Parentheses '(' and ')';
3) Operators "+", "-", "*", "/", and "^". The latter operator is that of exponentiation, which, has the highest precedence For the purpose of this project, the exponent can be only integer—so if, for example, 2^3.7 is entered, an error must be generated;

You should start by adapting the code in the book and then add the code to handle the exponentiation.

You will first need to parse an input (pre-fix) expression to convert it to post-fix notation. Your code must generate an appropriate error messages whenever it detects that the input expression is ill-formed.

Second, your code will evaluate the post-fix notation—just as we have discussed this in class (and as it is done in the book). Of course, the exponentiation operation is new, and it must be evaluated by repeating multiplication (and **not** by using Java native exponentiation). In other words, for the expression *a^n*, you explicitly need to produce the code that multiplies *a* by itself *n* times.

3   Submission instructions

Submit a single file named Expression.zip through Canvas that includes *PostfixEvaluator.java* and *PostfixEvaluatorTest.java* with your test cases. No report is required.

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) A non-integer exponent is allowed.

The code must be well-commented.

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