Assignment 3: Interpreter 101

1 Implementation

You are to implement an interpreter for a simple operator language in Java. The language is a subset of the Java expression language with integers and floats for primitive types. The interpreter is to evaluate the expressions in exactly the same way as a Java would.

You will implement a Java class `Interpreter` with a main method that reads a file in the given language from standard input (`System.in`) and prints the value to which the parsed expression evaluates to standard output using `System.out.println`.

If the execution fails, the Interpreter should print the line and column number of the failing operator (`beginLine` and `beginColum` fields of `NodeToken`) with a descriptive error message to standard error. The format of the message should be “LINE:COLUMN MESSAGE”.

If the file does not parse, the program should print “Parse error.” to standard-error (`System.err`).

2 Remarks

The given grammar accepts expressions like “++5” or “–8”. You are not expected to handle these, you can print “5”, “7”, “8” or even throw an exception – the testcases will simply not cover these special cases. Note that in a language which does support prefix and postfix increments and/or decrements, the grammar would usually contain special productions for these constructs (to make it easier for the compiler or interpreter to detect these patterns).

3 Submission

You must submit the implementations to your subversion repository to the directory `3351/$USER/P3/`. Include only the provided grammar, the Interpreter implementation and the provided build script. The files must be called

- `expressions.jj`

- `Makefile`
- `src/edu/du/cs/comp3351/p3/Interpreter.java`

You must check that the submitted code compiles by invoking `make`. Verify that the output of your program matches the expected output using your own testcases.