Interpreter Assignment #1: Expressions and Assignments

Issued: Tuesday, September 16
Due: Tuesday, September 30

Purpose

This assignment asks you to extend a small interpreter.

Source Code and Grammar

A Java implementation of the interpreter is at:

```
1  ~buff/classes/354/pub/ia1
```

The interpreter employs an ad-hoc scanner and a recursive-descent parser. The parser builds a strongly typed parse tree, which is then traversed and evaluated. A grammar for the source language is:

```
1  stmt  : assn ';'
2  assn  : id '=' expr
3  expr  : term addop expr
4        | term
5  term  : fact mulop term
6        | fact
7  fact  : id
8        | num
9        | '(' expr ')'
10 addop : '+'
11       | '-'
12 mulop : '*'
13       | '/'
```

The interpreter is an instance of the object-oriented design pattern named Interpreter(243), from the well-known "Gang of Four" textbook used in CS 472.

Assignment

There are several parts:

- Document the provided source code.
- Extend the scanner to support comments. Design your own form of comment.
- The grammar specifies that a program is exactly one assignment statement. However, `main` interprets each of its arguments as a separate program, trying to modify variable values, in the `Environment` object, accordingly. Change the interpreter so this works correctly. Eventually, we will change the grammar to allow a program to be multiple statements.
- Add a prefix unary minus operator. This is a grammatical change; simply allowing negative numbers is insufficient.
- The interpreter currently supports only integer values. Change it to instead support double values.
- Test your solution thoroughly. I have provided a simple regression tester, named `run`. Note that it uses the whole `prg` file as a single program. You cannot put multiple programs in a single `prg` file. Add tests to my rudimentary test suite. The quality of your suite will influence your grade.