Interpreter Assignment #1:
Expressions and Assignments

Issued: Tuesday, February 9
Due: Thursday, February 25

Purpose

This assignment asks you to extend a small interpreter.

Source Code and Grammar

A Java implementation of the interpreter is at:

```
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:

```
stmt  : assn ';'
assn  : id '=' expr
expr  : term addop expr
        | term
term  : fact mulop term
        | fact
fact  : id
        | num
        | '(' expr ')'
addop : '+'
        | '-'
mulop : '*'
        | '/'
```

You can compile and run the interpreter, like this:

```
javac *.java
java Interpreter "x = 1+2;" "y = x+3;"
```

Note that you must quote Bash metacharacters on the command line. The regression tester, described below, provides a better way to run the interpreter. Note further that this runs the interpreter on two complete source programs, according to the grammar.

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:

- Test the interpreter thoroughly. I have provided a simple regression tester, named `run`. Add tests to my rudimentary test suite. IF YOU DO NOT DO THIS, YOUR SUBMISSION WILL NOT BE GRADED.

- Finish documenting 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.