CS 280: Program #4

Spring 2016

Due May 3, by 11:55pm, via Moodle

For this assignment we are expanding on assignment 3. Now that we have parsed our language in assignment 3, we will actually execute the statements in the language for assignment 4.

The remainder of the semantic rules of the language must be enforced. Invalid combinations of types and operators must be detected. Your program must statically test the result of the parse for any semantic errors, print an error message, and stop the execution of the program.

If the program successfully parses its input, and finds no semantic errors, it can proceed to evaluate the statements in the language.

The most straightforward way to evaluate the statements in the language is to provide an evaluation function for every node in the parse tree. The evaluation function performs the necessary calculations and operations and returns a Value.

Executing the statements in the language involves:

- Handling declaration of variables; note that you should not permit a variable to be declared more than once.
- Printing a value after evaluating it in a print statement.
- Assigning a value to a variable whenever executing an assignment statement.

You may implement this assignment by overloading various operators on the Value class if that makes sense.

You may use the solutions from Assignment 3 as part of your assignment 4, or you may use your own implementation. You MUST submit all code needed to compile and execute your program.

Note that you should change your implementation from assignment 3. You should remove the printing of the various node counts from your assignment.

Your program should read the file whose name is passed as a command line argument, or the standard input if no command line argument is provided. You may divide this assignment into as many files as you like. You MUST use p2lex.h from the last assignment, with no changes. You MUST have your lexical analyzer in a separate file.