CSci 450: Org. of Programming Languages
CSci 503: Fundamental Concepts in Languages
Assignment #5, Fall 2018

H. Conrad Cunningham

17 November 2018

Assignment #5 (Optional)

Due 11:59 p.m., Friday, 30 November

General Instructions

All homework and programming exercises must be prepared in accordance with the instructions given in the Syllabus. Each assignment must be submitted to your instructor by its stated deadline.

Citations: In accordance with expected scholarly and academic standards, if you reference outside textbooks, reference books, articles, websites, etc., or discuss an assignment with individuals inside or outside the class, you must document these by including appropriate citations or comments at prominent places in your submission such as in the header of the primary source file.

Identification: Put your name, course name, and assignment number as comments in each file you submit.

Assignment Description

- This is an individual assignment.
- This is an *optional* assignment. It can replace a lower grade on one of the first four assignments.
- Read Chapters 42 and 43 of the ELIFP textbook if you have not already done so. (If you need additional information on the concrete syntax and parsing, examine Chapters 41 and 44.)

- All students: Do exercises 1, 4, and 5 from Calculator: Abstract Syntax & Evaluation (Chapter 42) of the ELIFF textbook.

  These exercises ask you to modify the **Prefix Syntax** interpreter for the ELI Calculator language by adding several new operations.

  I have already made the needed modifications to the abstract syntax module (i.e., `Expr` data type) and all other Prefix Syntax modules except the Evaluator module. I have added code for the `If` operation to the Evaluator module. (This was an exercise on Exam #3.) You will need to complete the other needed changes to the Evaluator module.

  The modified source code for the various Prefix Syntax modules are linked below:

  - Evaluator module `EvalCalc_hw05` (partially modified)
  - Abstract Syntax module `AbSynCalc_hw05`
  - Values module `Values` (no modification needed)
  - Environment module `Environments` (no modification needed)
  - Prefix parser module `ParsePrefixCalc_hw05`
  - Lexical Analyzer module `LexCalc_hw05`
  - Prefix Parser REPL module `PrefixExprREPL_hw05`

  Please use the same file and module names for your submission.

- Test your module thoroughly. In addition to any test programs, you may want to use the modified REPL `PrefixExprREPL_hw05` to test your modifications.

  The REPL is a Haskell `IO`, so you will need to call function `main` after starting the REPL.

- When complete, submit all seven Haskell modules and any additional testing or documentation files you use to the course Blackboard site for Assignment #5.

  Be sure to document your code appropriately using program comments. Give attention to the general instructions given above and in the Syllabus.