COS 301 Programming Languages
Fall 2012
Project Assignment #5 Due Last Class

Evaluation and Conclusion

For this assignment, submit the paper with your graded first draft of Paper 1 or Paper 2 that received the lowest grade! There will no rewrite cycle and your grade will be partly based on the degree of improvement shown since the start of the semester.

Write a paper (as many pages as you think are needed) that provides a critical evaluation of your project language (what is good and what is not) and a summary of what you have from learning a new language. Also describe, if applicable, how your experience with the language has changed how you think about programming and problem solving; or if not, why not. You might consider starting the paper by reconsidering and rewriting your language overview from paper 1.

Unlike earlier papers, this paper has no assigned specific technical content. You can of course include as much as needed to make your points and support your arguments.

Programming Assignment #5

An algorithm for converting infix to postfix expressions without implementing an expression grammar was designed by Edsger Djisktra and is known as Djikstra's Shunting Yard Algorithm. For this assignment, locate source code that implements the Shunting Yard Algorithm in your project language, adapt it to your needs; implement an RPN expression evaluator to processes the output (evaluate the expressions), and run and test the code using the following expressions:

a. x + y^z - d
b. (a + b) / (c + d) * e
c. x - j * c ^ (d - e) - 1
d. ((a - c) + (d - e)) / (d - c * a) + z

where a = 42, b = 5.1, c = 2, x = 1.22, d = 3.45, e = 5.45, j =3, y = 7.0 and z = 3.

You may have to substitute numeric values for variables and you many have to use another operator (or a function) for exponentiation. Djisktra's algorithm handles functions and their arguments in addition to simple expressions involving only binary operators and variables.

Searching Google for "shunting yard algorithm [your language] will likely yield usable code in the first five or ten results.

Submit (1) your complete source code; (2) a reference to the code you use or adapted; and (3) sample output.