Requirements Analysis and Implementation of a Symbol Table

Small Group Activity

Due: Start of class next Thursday, one week from exam. We will be discussing the solutions in that class period.

Instructions:

0. Create a group of 2-3 students to work on this project. You may try it on your own if you like.

1. Choose either COOL or C as your language of interest.

2. For your language, write a list of the scoping and access rules that the symbol table needs to implement and enable checking. Try to organize and group the rules in some way to make it easier to think about them all.

For example,

All method names in the same class must be unique.

Global variables declared outside the functions in a given file are visible from all functions declared in that file, but not visible outside the file.

Email a word file with your organized list of rules to Pollock@cis.udel.edu before class and bring 2 printouts per group to class.

3. Design a symbol table(s) data structure that you think will work well for keeping the information for each name in your language's programs. Describe the overall data structure maybe with a small example picture to make it clear. Write pseudocode for lookup, insert-new-name, enter-new-scope, and exit-scope operations. Note that if more than one item can have the same name, (in different scopes) then your symbol table has to have more than one entry for that same name. Hint: Look at your textbook or other compiler books for ideas.

Create a set of overhead slides to explain your data structure and the operations. Be prepared to present and justify your choices.

4. Go back to your list of rules, and for each rule, add a sentence describing how you would check that rule, or implement it with your symbol table. Try to combine as many rules being checked as possible during the same traversal of the abstract syntax tree and use of the symbol table.

Email the extended word file to Pollock@cis.udel.edu before next class.