

CS245 Lab 3: Lexical Scoping

Due Wednesday of the 7th week of classes (21 Oct 2009) handed out 5th Tuesday.

This lab, like labs 0 and 1, has several programming projects. When you are ready to start programming, check out the `Scoping` and `Scoping-HERA` projects.

You should do Question 1 on your own, but then are invited to work on Questions 2 and 3 as a group of up to four people, making sure that all group members' names are on all papers and files you submit.

1. Write a scheme function named `which-scope-rules`, in the file `which-scope-rules.scm`, to investigate the scoping rules of scheme, with respect to function parameters and/or local variables created by `let`. Your function should return the string “dynamic” if the scheme system running the function uses dynamic scoping and “static” if it uses static scoping.

You may use at most one global function in your answer to this lab, and no global variables. You will almost certainly need to combine several uses of `let` or local function definitions.

If you have good notes from class, this should be almost trivial.

2. The file `three-levels.scm` contains several scheme functions. Draw (on paper or with an editor such as inkscape or gimp or whatever you like, as long as you can produce a printout or a PDF file to submit) a memory layout diagram or memory evolution diagram (your choice) showing the execution of the `test` function up to the point in time at which the child function finally adds `x` and `z`.

3. Convert the Scheme functions from Question 2 (except for `print`) into HERA. Run both programs and confirm that they produce the same final result. Set a breakpoint in the debugger and capture an image of a window showing the program and the memory at the time at which the child function finally adds `x` and `z`.

Remember to submit your work with `Team->Commit` and handing in paper.