

CS350 Lab 7: Code Generation for Functions and Variables

Change the definition of `which_cs350_lab` in `tiger.cc` so that it has the value 7.

Add information to your symbol table so that each variable added to the symbol table has information about its offset within the frame and enough other information for you to be able to figure out which frame it came from. You may need to add other attributes to your AST to do this. Then add code generation for variables, loops, and function definitions, assuming that all variables are integers or strings. For the lab, you do not have to keep variables in registers, so you can treat every assignment as a `STORE` into the stack, and every variable use as a `LOAD` from the stack.

Record important design decisions for this lab (including what you add to the symbol table, and what new attributes you add to the AST) in the file `Lab7-Functions` in the `Design_Documents` subdirectory.

You do not need to handle arrays, records, or type declarations yet, as these are the subject of the next lab. For 70% partial credit, don't worry about non-local variables.

OPTIONAL: If you wish (and for a to-be-determined amount of bonus credit), you may improve on the basic system by either

- a) designing, documenting (in your design documents folder), and implementing a way to handle function calls that have other function calls in their parameter expressions; or
- b) designing, documenting (in your design documents folder), and implementing a way to pass parameters in registers rather than on the stack. You'll have to calculate "escape" information and handle parameters that escape). You will also have to handle functions that have more variables than there are HERA registers.

I strongly recommend that you do not attempt either optional step until you have the basic lab working, and that you do a "Team->Commit" before starting on optional work (as well as when you're done with the lab).