Book Assignment #2:
Names, Scopes, and Bindings

Issued: Tuesday, March 25
Due: Thursday, April 10

Purpose

This assignment asks you to think about the management of names in programming languages: scope (aka, visibilty), extent (aka, lifetime), and binding (i.e., meaning).

First, read the question in or textbook. Then, read what this handout has to say about the question.

3.2 This question asks you to differentiate static and stack allocation. Use Java and Scheme for your two examples, respectively.

3.4 This question asks you to differentiate lifetime and visibility (aka, extent and scope). Use Java and Scheme for your examples.

3.5 This question asks you to differentiate declaration-order rules, for different languages.

3.6 This question asks you to draw stack frames.

3.13 This question asks you to differentiate shallow and deep binding (aka, late and early binding), in dynamically-scoped languages.

The question asks "What does this program print?" Of course, it doesn't *print* anything. The real question is: What is the result of evaluating the expression `(A)`?

We discussed `define` and `lambda` during the lecture on Scheme. Recall that Scheme is actually statically scoped. In Scheme, the result of evaluating `(A)` is `2`.

3.19 This question asks you to differentiate static and dynamic scope *and* deep and shallow binding.