Today’s pair-programming partnerships are:

- Mr. Nordin and Ms. Kyaruzi
- Ms. DeWitt and Mr. Schlager
- Mr. Dewey and Ms. Jervis
- Mr. Mulhall and Mr. Champagne
- Ms. Liddell and Mr. Boateng Asante
- Mr. Mataire and Mr. Zahid
- Ms. Byrne

In each pair, the partner listed first should log in and manage the keyboard and screen.

**Exercises**

**A.** Create a subdirectory called `NAMELESS` in the directory you’re using for labs in this course and copy my implementation of the nameless interpreter for the PROC programming language into it. The `/home/stone/courses/languages/code/NAMELESS` directory contains eight language-specific modules:

- `expvals-and-environments.scm`
- `static-environments.scm`
- `tokens.scm`
- `syntax-trees.scm`
- `scanner.scm`
- `parser.scm`
- `translator.scm`
- `interpreter.scm`

You’ll also need `/home/stone/courses/languages/code/list-of.scm`, a copy of which should be placed in the parent directory of `NAMELESS`.

**B.** Consider this PROC program, from page 93 of the textbook:

```
let x = 37
in proc (y)
    let z = -(y,x)
    in -(x,y)
```

1. Submit this program to the translator and confirm that the result matches the one shown in the textbook. If it does not, explain the discrepancy.
2. Without running the translator again, predict how the translation will change if we change the `x` in the last line of the program to `z`.
3. What is the value of the program shown above? How does the value of the program change if we change the `x` in the last line of the program to `z`?

**C.** (1) Is it possible to apply the `value-of-program` procedure in the nameless interpreter to a syntax tree that would cause the `report-no-value-found` error (in the `expvals-and-environments` module) to occur? If so, suggest such a syntax tree; if not, show that there is no such syntax tree.
(2) Is it possible to apply the `translation-of-program` procedure in the nameless interpreter to a syntax tree that would cause it to return a translated syntax tree that, if provided as an argument to `value-of-program`, would cause the `report-no-value-found` error to occur? If so, suggest such a syntax tree; if not, show that there is no such syntax tree.

(3) Is it possible to apply the `scan&parse` procedure in the nameless interpreter to a string that would cause it to return a syntax tree that if provided as an argument to `translation-of-program`, would cause that procedure to return a syntax tree that would cause the `report-no-value-found` error to occur? If so, suggest such a string; if not, show that there is no such string.

**D.** A nameless interpreter for a version of PROC that supported multiple-parameter `proc`-expressions, multiple-argument call-expressions, and `let`-expressions that could introduce several local bindings simultaneously might use a ribcage representation of environments and lexical addresses with two coordinates — the first coordinate still indicating the number of scope boundaries lying between the use of a variable and its declaration, and the second indicating that variable’s zero-based position in the parameter list of the `proc`-expression or the list of bindings in the `let`-expression that declared it. Revise `expvals-and-environments.scm` so that lexical addresses are pairs of natural numbers rather than simple natural numbers.

**E (time permitting).** Draw up a list of passages in the implementation of the nameless interpreter that would have to be changed in order to accommodate multiple-parameter `proc`-expressions, multiple-argument call-expressions, and `let`-expressions with multiple binding specifications. In particular, consider how the translator and the `value-of` procedure in the interpreter proper would need to be changed.