

1 **CS370 Programming Languages (Zaring)**2 **Spring 2020**3 **Assignment 5**4 **Due by 11:59pm on Thursday, April 9, but accepted without late penalty until 4:00pm on**5 **Friday, April 10**6 **No assignments accepted after 4:00pm on Friday, April 10**

7

8 **Description:**9 (a) Extend the interpreter in `interp00.1.ss` so that it handles the following new  
10 *static-scope-named lambda-expressions*:

11

- 12 • `(lambda sym0 (sym1 ... symn) expr)`
- 13 `(lambda sym0 (sym1 ... symn . symn+1) expr)`
- 14 `(lambda sym0 sym1 expr)`

15 These expressions correspond to the existing lambda-expressions, with the addition of  
16 *sym<sub>0</sub>*. In each case *sym<sub>0</sub>*, serves as the “name” for that lambda expression’s static scope,  
17 which can be used with the following new expression.

- 18 • `(:: sym0 sym1)`

19 The value of this expression is the value of *sym<sub>1</sub>* in the static scope level introduced by the  
20 static-scope-named lambda-expression whose name was *sym<sub>0</sub>*.

21

22 For example, when run on the expression in the file `exampleA.rkt`, a correct  
23 implementation should produce the value

24

```
25 (seven (six) two (four2 four3 four4) three one one)
```

26

27 Save your extended interpreter in the file `assign05a.rkt`.

28

29 (b) Extend the interpreter in `interp00.1.ss` so that it handles the following new  
30 expressions:

31

- 32 • `(static sym)`  
33 Produces the same value as does just *sym*; that is, it produces the value bound to *sym* with  
34 respect to static-scoping behavior (which is Scheme’s usual behavior)
- 35 • `(dynamic sym)`  
36 Produces the value bound to *sym* with respect to dynamic-scoping behavior

37

38 For example, when run on the expression in the file `exampleB.rkt`, a correct  
39 implementation should produce the value

40

```
41 (b b a)
```

42

43 Save your extended interpreter in the file `assign05b.rkt`.

44

**45 Strategy:**

46 This is an exercise in modifying the aspects of the interpreter in `interp00.1.ss` that pertain  
47 to scope rules. Take your clue from the sorts of modifications that were made to produce the  
48 interpreters in the files `interp00.1d.ss`, `interp00.1l.ss`, and `interp00.1f.ss`.  
49 You won't need to write/modify all that much code for this assignment, but the  
50 changes/additions you'll need to make can be subtle.

51

52 As always, you'll be graded on program correctness, style (including choice of  
53 identifiers/symbols), and documentation (including the required header comment described at the  
54 beginning of this handout), just as you have been in your earlier computer science courses.

55

**56 What to Hand in:**

- 57 • Your files `assign05a.rkt` and `assign05b.rkt`, submitted using the Assignment 5  
58 item on the `Assignments` page of the CS370 Katie course