CSSE 304 Assignment 12 40 points

This problem has been used in the past, so you can probably find an old solution and "clone" it. **Don't do it!** You need to really do this problem (and perhaps several others) to be ready for the E&C part of Exam 2.

Turnin (Winter, 2019-20): Bring a written or printed (two-sided) solution to the box outside my office by noon on Wednesday.

No late days may be earned or used for this problem.

The actual problem is on the back of this page, and you should place your solution there.

I recommend doing the problem on scratch paper first, then copying it neatly to this paper.

You can print additional copies of this document as needed. To save paper, please print double-sided or just print page 2.

You can do this alone or with your project partner. If you do it together, you should submit only one paper with both of your names on it.

Your diagram must follow the style used in the class examples:

1. A closure has **three parts** (argument list, code, environment pointer).
2. A local environment has **two parts** (a table of variables and their values, and a pointer to an environment produced by enclosing code (if any).
3. Environment pointers always point to environments, never to closures or code.
4. The value associated with a variable in an environment is never an environment, and it is never code.
5. You can't have two arrows coming from the same "pointer location".
6. Arrows never point to something "inside the box" of an environment or closure.
7. Place sequence numbers (start with 1) near each environment or closure that you draw and near each new entry in the global environment, to indicate the order in which these references are created during the execution of the code.
8. For simplicity in the case of `let`, you should pretend that `let` is executed directly (without translation into an application of `lambda`), so that all you need to show is the environment extension, rather than creating a closure followed by the environment extension created by the application of that closure.
9. Show the changes this code makes to the global environment, and also include those in your sequence numbering.
10. **Hint:** My solution introduces 5 local environments, 4 closures, and 2 changes to the global environment.

Assignment 12. Name _____ Section (circle one) 10:00 11:00 12:00

(40 points) Draw a "environments and closures" diagram like the ones we did in class, showing all of the closures and local environments that are created by evaluation of the following code, as well as the relationships between them. Include sequence numbers for the creation of all closures and environments, as well as for additions to the global environment. Your diagrams must follow the style used in the class examples (see Page 1 for details):

```
(define compose2
  (lambda (f g)
    (lambda (x)
      (f (g x)))))

(define h
  (let ([g (lambda (x) (+ 1 x))]
        [f (lambda (y) (* 2 y))])
    (compose2 g f)))

(h 4)
```

Output from `(h 4)`: