

## 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.

Solve this problem after you have solved and checked the solutions for the warmup problems in the same directory.

**Turnin:** Submit a scanned version to the A12 drop box on Moodle OR slide a paper version of the assignment under my office door before the deadline. I request you submit this document as a PDF. Please put your name on top of your submission. You only need submit the second page.

You should do this problem alone, but after you are finished you may check it with your project teammate. Every team member should submit their own final version of the solution. If you do make a lot of changes after discussions with your teammate, please go ahead and re-print the second page and draw it from scratch.

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.

### The problem:

(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. Also show the final output of the code. 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.

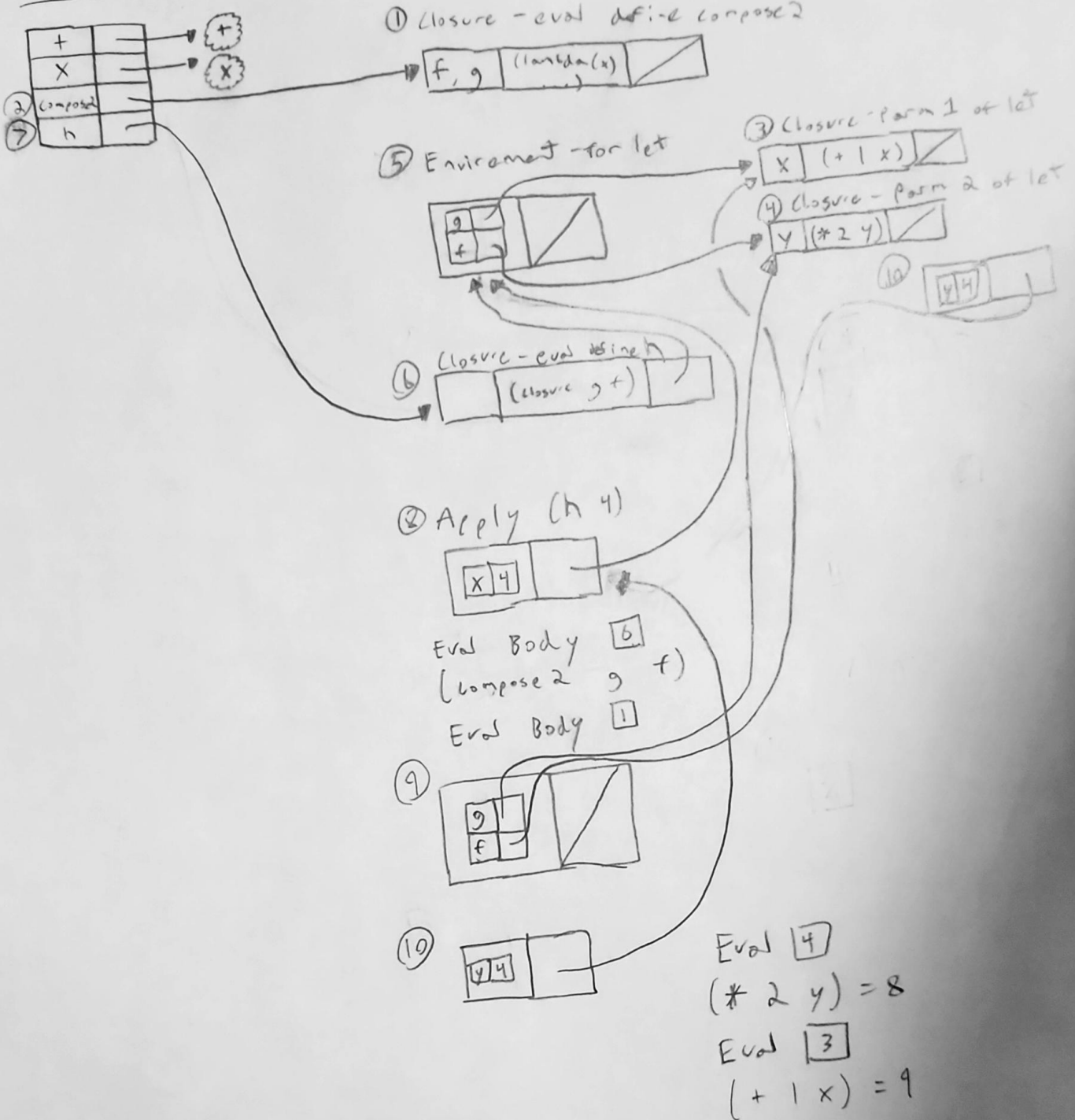
```
. (define compose2
  (lambda (f g)
    (lambda (x)
      (f (g x)))))

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

(h 4)
```

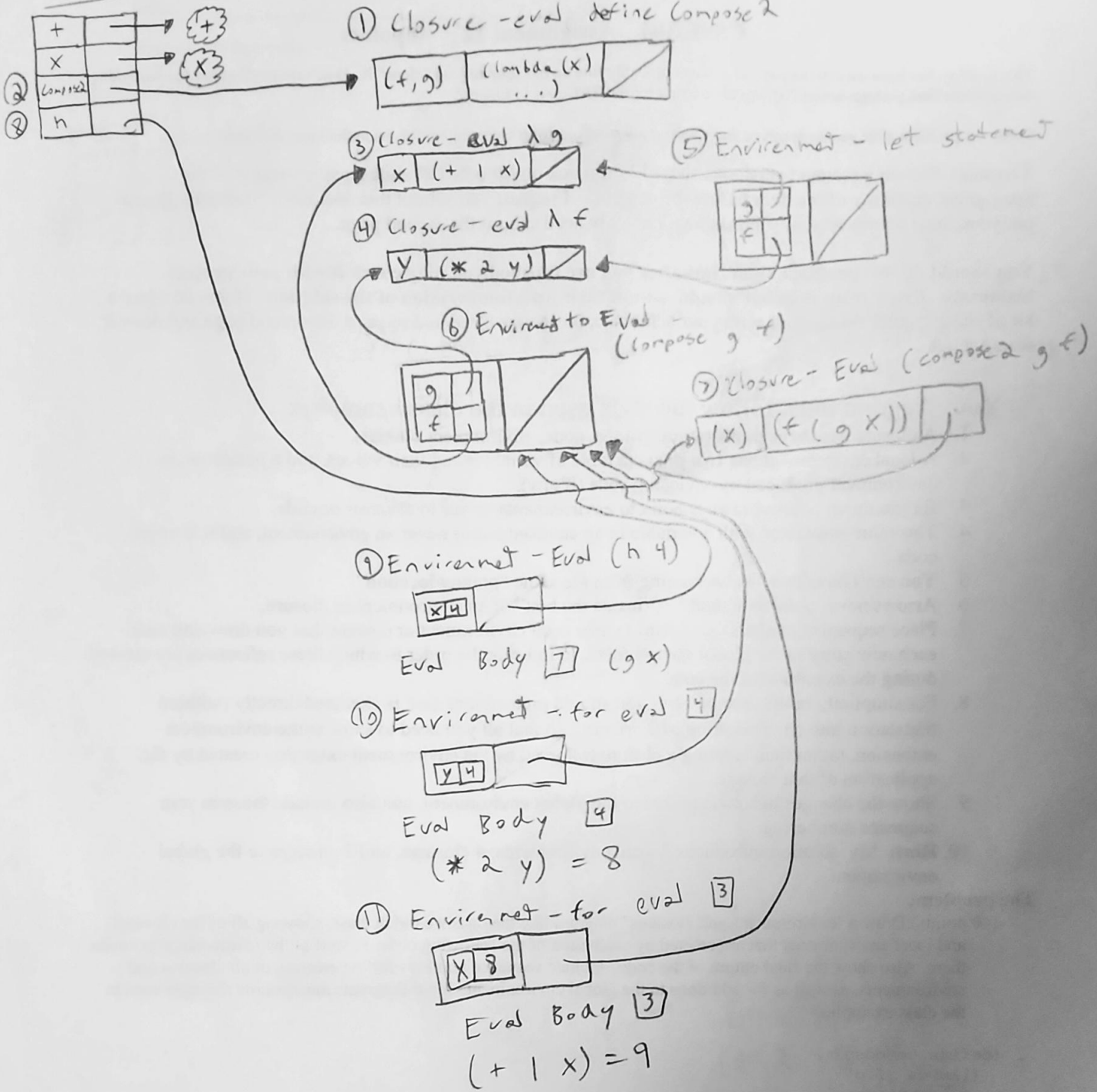
Draw your solution below.

Version 1



**Answer: 9**

# Version 2



Answer: 9