Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Environments and Closures Practice In Class

**DAY 1 (25 points)**

(let ([compose2 (lambda (f g)

(lambda (x)

(f (g x))))]

[a (lambda (y) (+ 1 y))]

[b (lambda (z) (\* 2 z))])

(let ([h (compose2 a b)])

(h 4)))

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Environments and Closures Practice In Class

**DAY 2 (25 points)**

(define my-odd (letrec ([odd? (lambda (x)

(if (zero? x)

#f

(even? (sub1 x))))]

[even? (lambda (y)

(if (zero? y)

#t

(odd? (sub1 y))))])

odd?))

(my-odd 2)