Quiz 6b Rubric

1. (3 points) Identify the programming style of the following code snippets.

Answers:

C. Message passing

A. Conventional

B. Data Directed

1 point, all or nothing, for each.

1. (4 points)

((lambda (x y) (+ 5 x y))

10

((lambda (x) (\* 2 x)) 1))

(1 points) a) What does the above code return? \_\_17\_\_\_

1 point for correct answer.

-0.5 if student used substitution method, showed work,

but messed up and got incorrect answer.

(3 points) b) How does scheme-1 evaluate the expression?

The first call to eval-1 is given. Show the next 3 calls to eval-1.

(eval-1 ‘((lambda (x y) (+ 5 x y))

10

((lambda (x) (\* 2 x)) 1))

(eval-1 ‘(lambda (x y) (+ 5 x y)))

(eval-1 10)

(eval-1 ‘((lambda (x) (\* 2 x)) 1))

1 point for each correct eval-1 expression.

-0.5 points for wrong order.

-0.5 for improper quotes

1. (3 points)Let’s build a **message-passing** model of a color!

make-color should take 3 arguments: red, green, blue

A color should respond to the following messages:

‘red : return the red value

‘green: return the green value

‘blue: return the blue value

‘intensity: return the average of red, green, and blue

Here is a sample interaction:

> (define c (make-color 12 1 2))

c

> (c ‘red)

12

> (c ‘green)

1

> (c ‘blue)

2

> (c ‘intensity)

5

Fill in the code to make this happen:

(define (make-color r g b)

(define (dispatch op)

(cond ((eq? op ‘red) r)

((eq? op ‘green) g)

((eq? op ‘blue) b)

((eq? op ‘intensity) (/ (+ r g b) 3))

(else (error “Unknown op”))))

dispatch)

3 points for functional message-passing code

2 points for everything but intensity

1 point for functional code but in wrong style