Quiz 0.2

1. (3 points) What will Scheme print? If it causes an error, write Error.
   1. (define (double x) (\* 2 ‘x))

(double 4)

Error\_\_\_\_\_\_\_\_\_\_

* 1. (define x 10)

(define (square x) (\* x x))

(square 3)

9\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. (define (mystery wd)

(sentence (first wd) (bf (bf wd))))

(mystery ‘hello)

(h llo)\_\_\_\_\_\_\_\_\_

Grading: 1 point each, all or nothing.

1. (3 points) Suppose we gave you a procedure month which took a number between 1 and 12 and returned the name of the corresponding month (as a word). Write a procedure dateformat which works like this:

> (dateformat ‘(8 29 2012))

(august 29 2012)

(define (dateformat date)

(se (month (first date)) (bf date)))

Grading: Take off 0.5 points for each minor mistake, with up to 1 point off for various types of errors (for example, don't take off more than 1 point for paren errors). Don't penalize them if they wrote month, but mention that they didn't have to write it.

1. (4 points) Consider what happens when you type (1 + 2) into the interpreter:
   1. How does Scheme try to evaluate this expression?
   2. At which point in the process does Scheme realize that there is an error?
   3. What is the rule about procedure application that this expression did not follow?

In order to evaluate (1 + 2), Scheme evaluates its subexpressions. It evaluates the 1 and the 2 to themselves, and evaluates + to the addition procedure. It then tries to invoke 1 as a function since it is the leftmost subexpression. At this point, an error occurs.

The rule is that the procedure being invoked must always be the first subexpression.

(Note: Should also accept that error occurs as soon as Scheme evaluates the 1, because it is not a procedure. That may be true in some implementations.)

Grading: 2 + 1 + 1 points. Don't take off points for incorrect terminology as long as their meaning is clear.