**Quiz 0.2b Rubric**

1. (2 points) Write these mathematical expressions in terms of a Scheme expression. For exponents, assume we give you a function called pow whose first argument is the base and second argument is the power. For example, (pow 3 2) is equivalent to 32 .
   1. 2\*(32 + (43 / 76 )

\_\_\_(\* 2 (+ (pow 3 2) (/ (pow 4 3) (pow 7 6))))\_\_\_\_\_\_

\_\_\_\_\_\_(/ (- 5 (\* 6 9)) (pow 2 (/ 3 7)))\_\_\_\_\_\_\_\_\_

**Rubric:** one point each

1. (2 points) What will Scheme print? If it causes an error, write error.
   1. > (define (proc x y) (/ (first x) (bf y)))

> (proc ‘62people ‘(5 6))

\_\_Error\_\_

* 1. > (define foo 5)

> (define bar 7)

> (define (foo bar) bar)

> (foo 6)

\_\_\_6\_\_\_

**Rubric:** 1 point each

1. (6 points) There are 3 things in this code that will cause an error. Specify the errors (be specific!) and explain why it is an error.

> (define (new-if pred cons alt)

(if pred

cons

alt))

> (if (#f)

(/ 4 0)

(new-if (+ 2 3)

(/ 4 0)

(+ 1 (bf ‘(1 0)))))

(#f) false is not recognized as a procedure.

new-if has to evaluate (/ 4 0) which results in an error, they should mention its the (/ 4 0) in the new-if, not the normal one

(+ 1 (bf ‘(1 0))) will cause an error since 1 can’t be added to ‘(0)

**Rubric:** 1 point for each error and 1 point for each explanation