Quiz 3, CSCI 360, Fall 2004

Name: _____

1. [7 pts] Give a lambda expression representing the *composition* function. That is, the function represented by your expression should take two functions f and g as arguments and return the composition of the two functions — i.e., a function h where $h(x) = f(g(x))$ for any x.

2. [10 pts] Suppose we are using an applied lambda calculus including an infix subtraction operator. Reduce the following using normal evaluation order, showing each intermediate step.

$$(\lambda a.\lambda b.\lambda c.a\ c\ c)\ (\lambda d.\lambda e.d - e)\ (2 - 3)\ (5 - 8)$$

3. [10 pts] Write a Scheme function `MAP` that takes a function and a list of elements and returns a new list where each element is the result of applying the parameter function to the corresponding element in the parameter list. For example:

`(MAP (lambda (x) (* x x)) '(4 7 8))` returns `(16 49 64)`

In this case, the programmer has passed in the squaring function, and the `MAP` function creates a list holding the squares of the original list.

4. [5 pts] Which of the parameters of the following Haskell function are strict?

```
mystery f b y = if b then y else f y
```

5. [10 pts] Write a Haskell function *threshold* that takes a function f and a number z and returns a new function g for which

$$g(x) = \begin{cases} f(x) & \text{if } x < z \\ 0 & \text{otherwise} \end{cases}$$

6. [8 pts] Using Haskell syntax for writing types, give the most general type for the following function.

```
applyIfDifferent f x y = if x == y then f x x else f x y
```