Homework 1
Due Friday, 04/08/11

Please turn in your written solutions along with printouts and test runs of any programs in class. Also turn in any programs online via Blackboard. Both are due before the beginning of class on the due date.

1. (10 points) **Partial and Total Functions**
   Please do problem 2.1 from Mitchell, page 16.

2. (15 points) **Halting Problem on No Input**
   Please do problem 2.2 from Mitchell, page 17.

3. (20 points) **Conditional expressions in Lisp**
   Please do problem 3.2 from Mitchell, page 40.

4. (10 points) **Deep reverse**
   Write a Scheme function that will do a deep reverse of a list structure. That is, it will reverse not only the top-level list, but every sublist, sub-sublist, etc. Thus calling:

   ```scheme
   (deepReverse '(A B (C (D E F) (G H I)) J) (((K L))) M))
   ```

   should give:

   ```scheme
   (M (((L K))) (J (((I H G) (F E) D) C) B A))
   ```

   You should use a combination of recursion and a higher-order function. You may use the reverse function.

5. (25 points) **Lisp Interpreter Modifications**
   Add the functions `and`, `or`, and `not` to the Lisp interpreter that we saw in class. The functions `and` and `or` can have an arbitrary number of boolean expressions and should use shortcut evaluation (evaluate boolean expressions in the order given, and quit when you know the answer). Do not use the `and` and `or` functions defined by Scheme in your program. Think carefully about where each modification belongs. You should include a run demonstrating that your modifications work correctly.

6. (10 points) **Lisp Interpreter**
   Consider the following Lisp expression:

   ```lisp
   ((lambda (x)
       (let ((fn1 (lambda (y) (+ x y))))
         (let ((x 5))
           (fn1 x)))) 3)
   ```
What does this expression evaluate to when given to the DrScheme interpreter? What does it evaluate to when given to the Lisp interpreter from class (using myeval)? Explain how each got the value that it returns. (We will study this situation in greater detail later in the course, and will see how to fix the Lisp interpreter so that it behaves in the same way as the DrScheme interpreter.)