COMP 210, Spring 2002, Homework 4
Due Friday, February 15, 2002 at the start of class

Before you start the homework, you should remind yourself of our General Advice, Advice on Homeworks, and Grading Guidelines. All are available from the class web site (http://www.owlnet.rice.edu/~comp210).

For this assignment, you should follow all the steps of the design methodology and include the results of each step as comments or code in the final materials that you submit. (For example, write your template as a comment—at the appropriate point in the development sequence—and copy it over when you fill it in.)

1. (2 pts) Consider the domain of natural numbers, as defined in Lecture on Wednesday (notes are on-line). Write a program **multiply** that takes two natural numbers and returns their product. Your program may not use the built-in multiply function; instead, you should use addition and subtraction to compute the answer.

   Show all the steps in the design methodology. Hand evaluate two cases. Use DrScheme to evaluate them, as well as other test cases.

2. (4 pts) We can define a **list-of-list-or-symbol** as

```
;; a list-of-list-or-symbol is one of
;;        – empty, or
;;        – (cons f r)
;;        where f and r are both list-of-list-or-symbol
;;        – (cons f r)
;;        where f is a symbol and r is a list-of-list-or-symbol

;; example data
   (cons 'fee
         (cons
               (cons 'fie
                        (cons  'foe  empty) )
               (cons 'fum  empty)
          ) )
```

   a) Write a program **symbol-count** that takes a list-of-list-or-symbol and returns the number of symbols occurring in the input list. For the example data given earlier, **symbol-count** would produce 4.

b) Write a program **flatten** that consumes a list-of-list-or-symbol and produces a new list-of-symbol that has all the symbols from the list-of-list-or-symbol, in their order of appearance. For the example data given earlier, the program would produce

```
(cons 'fee
     (cons 'fie
          (cons 'foe
              (cons 'fum   empty) ) ) )
```

3. (4 pts) Given the definition for an ftn developed in class Wednesday – that is

```
;; a ftn (family-tree-node) is either
;;   – empty, or
;;  – (make-ftn  name year  eye-color  mother  father)
;; where name and eye-color are symbols, year is a number, and
;;     mother and father are ftns
(define-struct  ftn  (name  year  eye-color  mother  father))
```

Develop a program **average-age** that consumes an **ftn** and returns a **number**, where the number is equal to the average age of the people whose records are in the family tree. You may assume that this program will only be used in 2002 and that all people listed in a given year were born on January 1 of that year.

[The latter assumption simply allows you to ignore the months and days. The **ftn** definition only holds a year. Thus, someone listed with year 1980 is counted as 22 years old.]