COMP 210, Spring 2000, Homework 3
Due Wednesday, February 23, 2000 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.

This is the *half-homework*–-a shorter assignment intended to get us back on the Wednesday to Wednesday schedule. Thus, it has fewer questions than a normal homework and is worth 5 points instead of 10 points.

1. Consider the following definition for a binary tree. (Binary trees are related to the binary search trees discussed in lab lecture four this week, except that they have no ordering constraint.) We can define a binary tree as

```
;; a btn (binary tree node) is either
;;         — empty, or
;;         — (make-btn  fact  left-child  right-child)
;; where fact is a symbol and left-child and right-child are btns
(define-struct  btn  (fact left-child right-child))
```

Develop the following Scheme programs.

a. (2 pts) In a binary tree, a node is considered a *leaf* if it has no children. Develop a program **count-leaves** that consumes a binary tree node and produces a number that corresponds to the number of leaves in the tree rooted at that binary tree node.

b. (2 pts) In a binary tree, a node is considered an *interior node* if it is not a leaf. Develop a program **count-interior-nodes** that consumes a binary tree node and produces a number that corresponds to the number of interior nodes in the tree rooted at that binary tree node.

c. (1 pt) Develop a program **count-nodes** that consumes a binary tree node and produces a number that corresponds to the number of nodes in the tree rooted at that binary tree node.