CISC 280 Spring 2007 Lab 9

Design a program **printTree** that behaves as below. Use the binary tree constructors and accessors from the text. Maintain data abstraction. Demonstrate your program on a variety of full and partially full trees.

```
>
   (define t1 (make-tree 5
                         (make-tree 2 (make-tree 1 () ()) ())
                         (make-tree 7
                                     (make-tree 6 () ())
                                     (make-tree 8 () ())))
)
 (printTree t1)
>
           8
     7
           6
5
     2
           1
()
>
```

Hint: Use the (display "spam") and (newline) functions. Write a simple helper function that prints a desired number of spaces. If you don't immediately see how to draw the tree above, try doing the one below first. Once you've got it, just re-order the steps of your procedure to get the behavior above (why does this work?).

```
> (printAltTree t1)
5
2
1
7
6
8
()
>
```

If you need to submit (see syllabus if you are unsure) submit your code file(s) and a script of several well-chosen test cases via MyCourses (due Thursday midnight) and on paper (to your TA's mailbox Friday by 1 p.m.) to receive full credit.

When you use MyCourses, remember that you can "upload" files multiple times, but you only click "submit" once.