HW 4 – Algebraic Datatypes
CS 421 – Fall 2012
Revision 1.1

Assigned Tuesday, September 25, 2012
Due Tuesday, October 2, 2012, 11:59pm
Extension 48 hours (20% penalty)

1 Change Log

1.0 Initial Release.

1.1 Changed all "buying" references to "selling"; Customer level datatype should include "Common"; Further clarified `getLevel` and `updateLevels`; adjusted text slightly to match these changes.

2 Objectives and Background

The purpose of this HW is to test your understanding of:

- Algebraic datatype system in OCaml

3 What to `handin`

Answer each problem below, save your work as a ML file, and hand in the hw4.ml file.

4 Problem: CS421 Ltd.

Make sure to put comments before each section of this problem to describe the semantics of the problem and the way you want to implement the semantics of the problem.

Be careful about the description of the problem. You should fully understand the description of the datatype we want you to implement and your implementation of the datatype should be exactly the same as the semantics of the description.

4.1 Customer Participation

CS421 Ltd., a company based in Urbana, Ill., has developed "healthy air", a new kind of product. The air is a bio-natural organic compound in otherwise sterile air, which the company claims can prolong people's life for 50% according to their tests. You are employed directly by the VP of Marketing, who has asked you to implement some programs to help with CS421 Ltd.'s marketing vision of a multi-level marketing strategy through home sales.

Multi-level marketing is a business strategy providing the chance for customers to earn money through their own purchases. Because the customers who like our products tend to recommend our products to other known friends, we would like to provide incentives for them to join in the sales system and earn money by inviting new customers with discounts.

First, CS421 Ltd. needs to classify its "enthusiastic" customers with different levels. This class of customers are very good at spreading the word and generate sales to lots of other people for the company, as compared to the other "common" customers who show up randomly and invite fewer or no numbers of people. Management has determined that the company should distinguish between customers with **five** different levels, four of which describe these "enthusiastic" customers based on their selling power.

(For the purposes of this company, "sells" means any sales generated by the customer or *any* of their customers they have invited, customers the invited customers invited, etc.).

- **Leader.** The highest level customer. Anyone who sells at least 400 cans of healthy air per month (the price of the healthy air is $1000/can) gets a 60% discount on their purchases from the company.

- **Expert.** The second highest level customer. Anyone who sells at least 150 cans of healthy air per month gets a 40% discount on their purchases.

- **Establisher.** A middle participating customer. Anyone who sells at least 60 cans of healthy air per month gets 20% discount on their purchases.

- **Starter.** Lower-level customer. Anyone who sells at least 12 cans of healthy air per month gets a 5% discount on their purchases.

- **Common.** Anyone who is not classified as at least a Starter is viewed as a common customer, and receives no discount.

1. **(4 pts)** Write an OCaml algebraic data type `level` to describe the five kinds of customers listed above.

2. **(4 pts)** Write an OCaml function `getLevel: (int -> level)` to take the total number of air cans a customer sells in a current month as input, then return the level the customer is for that month. (Assume the integer input is a whole number.)

3. **(4 pts)** Write an OCaml function `getDiscount: (level -> float)` to take the level of a customer as input and return the discount the customer should have.

4.2 Marketing

The VP of Marketing comes back to you, and would like you to help them write a program for their upper management to keep track of all its customer data. Since the system follows a multi-level marketing strategy, the company would like you to implement the management system following a tree data structure. The tree should differentiate between the two classes of customers: the four-level, enthusiastic ones described in the previous task, and the remaining common customers. Both of these two kinds of customers should contain customer's name (assuming it is unique) and the total cans the customer bought for themselves. The enthusiastic customers should also contain their customer level and a list of other customers who bought cans from this customer.

4. **(4 pts)** Write an OCaml type `managementTree` to describe the data structure the company wants above.

5. **(4 pts)** Write `updateSales ((string * int) list -> managementTree -> managementTree)`, an OCaml function to update the number of cans a customer bought in the current month; the customer name and the new total number of cans are the contents of the input list.

6. **(4 pts)** Write an OCaml function `updateLevels (managementTree -> managementTree)` to adjust the customers' levels in the input managementTree, based on the sales numbers in the tree. Remember that customers can count all sales of all of the customers underneath them (recursively!). (Assume all integer inputs are whole numbers.)