•Lab 4: Chp3&4

Problem A: *Chicago Car Wholesale, Inc .* Objectives: Using Decision Structures, Decimal Format class, and appropriate testing.

This company specializes in selling fleet cars to both private companies and public agencies/non-profits in Illinois and the surrounding states. Orders are made for a single car type. Public agencies and non-profits do not pay sales tabx, but others must pay a 6.25% tax. All orders receive an additional discount of 5% for orders more than 10 cars and 10% for orders more than 20 cars.

Create a class that represents an Car Order with attributes for buyer, car type, cost, quantity, and tax status. The last attribute will be of boolean type. In addition to the gets and sets for each attribute, make a constructor that accepts all attributes when creating the object as well as a default constructor; a method that returns the cost of the cars with discounts applied; and a method that returns the amount of taxes based on the cost of the cars. (Hint, this last method should use the previous method as a support method.)

Create a driver program that makes Car Order objects from the following data. Your driver will read at least two of the objects' data items from the keyboard and the other objects may use data items defined in the Driver program as program constants. The Driver should create a report (sample below) listing the buyer, the car type, the cost per car, the quantity, the discounted cost, sales taxes if any, and the total cost. Use the Decimal Format class to format the output of Double data types.

Use the following data for the execution you turn in:

Buyer	Car	Cost	Quantity	Tax Status
Speedy Rental	Mini Cooper	$22,150	15	Y
Zip Car Inc.	Ford Fusion	$27,495	6	Y
State Motor Pool	Toyota Sienna	$28,295	10	N
IT Delivery Co.	Sprinter Van	$32,380	25	Y

```
    Chicago Car Wholesalers
      Oct. 30th, 2012
    New Car Order Report

    Buyer              Car                 Cost per Car   Quantity       Cost
Sales Tax       Total Cost
    Speedy Rental       Mini Cooper        $22,150         15            $315,637.50
19,727.34      $335,364.84 ....
```

Problem B: *Lane's Coffee Kiosk.* Objectives: Using Decision Structures, appropriate testing, and formatted output.

The Coffee Kiosk needs to track the amount of weekly wages paid to the employees. Employees are classified as Barista's or Supervisor's. Barista's are paid for overtime (above 40hrs) at 1.5 times their regular pay. Supervisors do not get overtime pay but receive a $500 bonus plus their regular pay when the weekly sales are over $22,000. Make an Employee class which has attributes for name, payrate, and employee code (B=barista, S=supervisor) and get and set methods for each attribute. You will input these attributes from the keyboard. (Hint: use the CharAt() method to extract the employee code character from the keyboard input string.) In addition, make a method that calculates and returns the employee's weekly pay given their hours & total sales for the week. (Hint: data can be sent to and returned from the same method). Create 2 baristas with the following wages: Allen: $8.75/hr, Pat: $9.25/hr, plus 1 supervisor Chris: $10.50/hr. Next, read the week's data containing the employee's hours and total sales, then create a report showing weekly wages for each person and the week's total wages. *Use the following data for the execution your turn in.*

Week1: Total Sales = $24,600, Allen: 47hrs, Pat: 24hrs, Chris: 52 hrs

Week2: Total Sales = $17,900, Allen: 17hrs, Pat: 51hrs, Chris: 72hrs

Week3: Total Sales = $32,100, Allen: 40hrs, Pat: 38hrs, Chris: 63hrs

Your Report output should look similar to this:

```
Week1 Total Sales: $xx,xxx.xx
  Allen          $xxx.xx
  Pat            $xxx.xx
  Chris          $xxx.xx
  Total Wages:  $x,xxx.xx

Week2 Total Sales: $xx,xxx.xx
  James          $xxx.xx
  Pat            $xxx.xx
  Chris          $xxx.xx
  Total Wages:  $x,xxx.xx
```