CS 123/231 Introduction to Software Engineering 1st Sem 2004-2005
Programming Assignment #2
Due: Thursday, 23 September, 11:59pm

GROCERY SALES AND INVENTORY

Write a program that simulates sales activity and inventory monitoring in a grocery store. The grocery keeps several products in stock. Customers walk in and purchase products through counters that support barcode scanning. Shopper cards (also barcoded) are issued to customers who wish to avail of special discounts for selected products.

There will be three input files for this program, described below:

1) Customer shopper card data (*cards.txt*): This file contains information on all customers who have shopper cards. Each line in this file represents one such customer and consists of a customer name and a barcode (separated by a comma) through which the customer is identified. The customer name is the string of characters that starts with the first character in the line and ends with the character that precedes the comma. The barcode is a string of 12 digits.
2) Product data (*products.txt*): This file contains information on all products that the grocery carries. Each line in the file represents a product and consists of a product name, UPC barcode (two strings of five digits each separated by a space), unit retail price, a discounted price for card holders, and an initial stock level. These five fields are separated by commas.
3) Purchase data (*buy.txt*): This file contains order lines. Each line has the format: **begin <counter-number> <customer-barcode>, <UPC> <qty>, <UPC> <qty>, …, end** . The line describes the details of a purchase: the counter where the purchase is made, the customer's barcode if available (this field will be an 'X' if the customer has no shopper card), and the products purchased in specified quantities.

Note that the barcode data present in *buy.txt* may be presented in reverse (the orientation of a barcode may not be upright when it is scanned). For example, 23456 89345 is a valid scan for the product whose barcode is 54398 65432. The same notion holds for shopper card barcodes.

The program will read the first two input files to define which products and customer cards are valid for the grocery. It then processes the purchases encoded in *buy.txt.* The program should produce the following output files:

1) Counter activity summary (*activity.txt*): For each purchase line, print the counter number, customer name (if available), and the total price for that purchase.
2) Product inventory data (*levels.txt*): For each product, print the product name and the stock level for that product after all purchases has been processed.
3) Sales summary (*sales.txt*): For each counter, print the total sales, total discount given, total purchases, total number of purchases, and the number of anonymous (no shopper card) purchases. Corresponding grand totals are also printed.

4) Customer activity summary (*custs.txt*): For each shopper card holder, list the customer's name and all the names of the products that the shopper card holder has bought at least once.

All amounts and prices in the input and output files are prefixed with a P sign and are printed using 2 decimal places. Customer and product names may have spaces in between them.

The C++ project should be named "GrocerySimulation". You are required to include a class called "Grocery" whose constructor includes an integer argument *n* that specifies the number of counters for the grocery. This determines what counter numbers (1 to *n*) are valid for the grocery. In the main program, a single Grocery object should be created. The program then runs methods on that object to read input files and produce output files.

You will provide a UML Class Diagram that specifies all the classes included in your system, their relationships, and the essential methods and attributes for each class. **You may work in pairs for this assignment, but you should specify which student is responsible for which classes. Include this division of work in the class diagram (Enumerate the classes for each student). Points will be taken off from both students if you do not clearly specify this division.**

Submit a zipped file of the class diagram and the folder containing your C++ project (make sure to delete the Debug directory before zipping).

Keep posted on the course website for announcements and clarifications pertaining to this assignment. Sample files I/O will be posted on the website. A demonstration of the working system may be required.