**Object-oriented programming(c++) course project**

*This project is the final project of object-oriented programming (c++), and must be fulfilled independently.*

Deadline: June 8th, 2014

**Project Description：**

Design and realize a simple Supermarket shopping and checkout terminal system.

The system provides command line user interface, and users in this system are the cashiers in supermarket aisles.

**Functional Requirements：**

1. **Data importing（Prepare for shopping and checkout）**
   1. Supermarket goods data should be imported from a text file, containing the commodity code (unique), name, origin, and benchmark price for every good.
   2. Member information should be imported from a text file, containing the membership number (unique), name, gender, contact phone number, member level, and member points. Member level includes the gold card members, silver card members, and regular members.
   3. Shopping card information should be imported from a text file, containing the id(unique) and balance of all the shopping cards from this supermarket.
   4. The data format of above three text files of a text file must be designed by your own. Project documents submitted shall make an explicit description of these data format, supplemented by a set of sample conforms to the format (each file should contain at least 100 data records).
2. **Shopping and Checkout**
   1. User enters checkout command, and the system goes to the checkout mode.
   2. In checkout mode, user enters the commodity code and number (by default is 1), the screen should displays its name, origin and benchmark price, and the system should add it to the shopping list of current customer.
   3. If user enters command to close checkout, the system should exit checkout mode, and start payment mode at the same time.
   4. System supports three payment approaches, cash, bank card, the supermarket shopping card. For cash, the system should provide the change computing services. To do bank card payment, the user should input bank card number, name, and amount of consumption. To use supermarket shopping card payment, the user should input gift card number and consumption amount. The system should check the shopping card record data to determine whether the balance is enough, and update the data record (make deductions). Each shopping list checkout can be completed by a variety of payment combinations, and each payment can be used for many times. For example, in the checkout process of a shopping experience, customer can use multiple bank cards and shopping cards and specify different consumption amount.
   5. After payment, the system should record the shopping list and payment information in system and output to text file. At the same time the system should show it on screen.
3. **Membership Service**
   1. Before checkout, the customer could display his member card, and the screen would show the member’ information after user enters the member number. The system would offer discounts and add member points according to member level. Details are as follows:

|  |  |  |
| --- | --- | --- |
| **Member Level** | **Discount Rule** | **Member Point Rule** |
| **Golden member** | Every product should be discounted separately: No discount for goods lower than 10.00 RMB; 5% off for price between 10.00RMB and 20RMB; 10% off for goods higher than 20.00RMB. | Calculate points for the entire shopping list：For the consumption part below 1000 RMB, add 1.5 points for each yuan; for the part between 1000 RMB and 5000 RMB, add 1.8 points for each yuan; for consumption part over 5000 RMB, add 2 points for each yuan |
| **Silver member** | Execute comprehensive discount to the entire shopping list price：5% off for the part below 1000 RMB; 10% off for the part between 1000RMB and 2000 RMB; 15% off for the part over 2000 RMB. | Calculate points for the entire shopping list price：For the consumption part below 2000 RMB, add 1.2 points for each yuan; for the part between 2000 RMB and 5000 RMB, add 1.5 points for each yuan; for consumption part over 5000 RMB, add 1.8 points for each yuan |
| **Regular member** | Execute comprehensive discount to the entire shopping list price：no discount for the part below 2000 RMB; 5% off for the part between 2000RMB and 5000 RMB; 10% off for the part over 5000 RMB. | Calculate points for the entire shopping list price：Add 0.5 point for each yuan; |

* 1. After each purchase, the system would check the total points of current member. Ordinary member would be upgraded to silver member when points reach 10000, and silver member would be upgraded to gold member when points reach 50000.
  2. As the system may need to support more member types, special discount rules and points rules in the future, the designer should take into account system extensibility, and make use of the characteristics of object-oriented programming.

1. **Data Update**

The system provides the update functionality for member information and shopping card information update functionality. After user issues commands, the system real-time records the latest dataset and writes back a text file to disk (the user selects to overwrite the original input file or separately creates a new file).

**Project Document:**

1. To write a project document which including the following contents at least.
2. Name, student number, contact number, email and other necessary personal information
3. To provide the description about all function of the program, how to use all function and the description about data format of the input file
4. The usage scenarios of use cases (the screenshot about running program and text description can be include)
5. The description about architecture design and function of each module
6. The simple introduction of the concept of program design and development process
7. Other things should be described (e.g. technical details)
8. Project documentation using DOC/DOCX or PDF file format and the file size is no more than 10MB. Page size is A4 and documentation does not be over 20 pages.

**Notes：**

1. Proper annotations written in the program code, in order to improve the readability of the program。
2. Coding with standard C++ and can (and encouraged) use containers and algorithms provided by C++ standard library.

**Additional requirements for submitting a project：**

1. Create a folder named by student number and student’s name. Student number and name are divided by underline. e.g. 123456\_MyName\_C++\_Project.
2. Create a text file (readme.txt) in the above folder and indicate student number, name, contact number, email address and other necessary things. Create 3 subfolders (named Source, Document and Sample) to put code, documents and sample input file.
3. All the code must be put in Source folder and this folder can only be put code files, and may not be put input and output files, executable file or any other type of files. All code files should be placed in the same folder, just for CPP or HPP. The file which contains main() should be named Main.cpp. All the beginning of the code file should be written in the form of annotation student number, name, the filename and file description.
4. To compress the folder containing all of the submissions into a ZIP file (filename format : 123456\_MyName\_C++\_Project.zip), Sending it to 10.60.41.1 personal C++ directory before deadline.