SEHH2042 Computer Programming
Group Project – Gift Redemption System
(Due: 23:59, 28 Apr 2024, Sunday)

Expected Learning Outcomes

- develop computer programs in one or more high level language programming environment;
- design and develop structured and documented computer programs;
- explain the fundamentals of object-oriented programming and apply it in computer program development;
- integrate the computer programming techniques to solve practical problems.

Introduction

In this assignment, you are going to develop a "Gift Redemption System" that runs in the command line environment. The system maintains the records of customers and gifts which are available for redemption by customers using the shopping points called *CC points*. The system allows customers to make queries on the available gifts for redemption and check the CC points balance after using CC points on gift redemption.

Tasks

- Each group is required to write a Win32 Console Application program called **GRS.cpp**.
- Each student is required to submit a video recording (at most 2-minute long) to demonstrate his/her individual contribution in the group project.
- Each student is required to submit a Peer-to-Peer evaluation form (through the given Word file) via Blackboard.

Program Requirements

R0 When the program starts, the console should display a welcome message, followed by the Main Menu of the program. Users can enter the options of the corresponding actions (see **R1** to **R6** below).

```
Welcome Message designed by your group
*** Main Menu ***
[1] Load Starting Data
[2] Show Records
[3] Edit Customers
[4] Enter Customer View
```

```
[5] Show Transaction History
[6] Credits and Exit
*****************
Option (1 - 6):
```

R1 **[1] Load Starting Data**

When the user inputs 1 in the Main Menu, the system is loaded with starting data. The starting data includes records of (1) gifts, and (2) customers as shown in R1.1 below. After the starting data is loaded, the system returns to the Main Menu.

R1.1 The starting data to be loaded, together with the required data format of the data fields, are described below.

Gift records

Gift ID	Gift Description	Price (HKD)	Points Required
A01	LG Internet TV	3900	19000
A02	Pioneer Hifi Set	2400	11500
A03	Sony DVD Player	400	2000
B01	Healthy Air Fryer	1500	7300
B02	Tefal Microwave Oven	480	2400
B03	Famous Coffee Maker	1800	8800
B04	Smart Rice Cooker	600	2900
B05	TechCook Toaster Oven	450	2250
C01	Wellcome $50 Coupon	50	250
C02	Mannings $50 Coupon	50	250
C03	Carol Restaurant $100 Coupon	100	500
C04	Shell $200 Coupon	200	960
D01	Clever Headset	350	1750
D02	HP Optic Mouse	250	1250
D03	Stylish Bluetooth Speaker	800	3900

Data format of each field:

- **Gift ID**: A string starting with a letter and then two digits. The starting letter represents the gift category:

Starting letter in Gift ID	Gift Category
A	Audio & Video
B	Kitchenware
C	Coupons
D	Computer Accessories

- **Gift Description**: A string that may contain white space (assume at most 100 characters long)
- **Price**: An integer, the price of the gift
- **Points Required**: An integer, the CC points required to redeem the gift

Customer records

Customer ID	Rank	Points Balance
Tommy2015	B	8500
DavidChan	B	22800
Luna123	B	650
TigerMan	B	14000
Max5678	S	2580
Neo2000	S	8000
CCTang	S	33554
EchoWong	G	8650
Rubychow	G	28000
Ivy2023	G	12340

Data format of each field:

- **Customer ID**: A string that uniquely identifies a customer in the system (assume at most 50 characters long); it is **case-sensitive**. You can assume it does not contain white space.
- **Rank**: A character (G, S or B) to represent the rank of the customer; there are different redemption discount policies for customers under different ranks, see requirement R7. Customers are ranked according to the length of time they become the member:

Rank (character)	Rank (description)	Become member for …
G	Gold	More than or equal to 1 year
S	Silver	Less than 1 year but more than or equal to 6 months
B	Bronze	Less than 6 months

- **Points Balance**: The CC points balance owned by the customer

R1.2 Options 2 to 5 in the Main Menu are enabled only after the system is loaded with the starting data. If the user enters options 2 to 5 before starting data is loaded, an error message should be shown, and then the system returns to the Main Menu.

R1.3 After the system is loaded with the starting data, the gift records data **CANNOT** be updated by any operations. However, the customer records data can be edited under options 3 or 4 in the Main Menu. See the requirements that follow.

~~R2~~ [2] Show Records

[After the starting data is loaded] When the user inputs 2 in the Main Menu, the system displays all the fields of all gift records, and then all the fields of all customer records. Tabular format should be displayed, and the records are **sorted in alphabetical order** based on the ID values. Data shown should be the **latest** set of data since the starting data is loaded (updates on the customer records resulted by the operations under options 3 or 4 in the Main Menu should be included). After showing the records, the system returns to the Main Menu.

~~R3~~ [3] Edit Customers

[After the starting data is loaded] When the user inputs 3 in the Main Menu, the system prompts for the next user input of a customer ID. If such customer ID does not exist in the system, it is an *add customer* operation. Otherwise, it is a *delete customer* operation.

Add customer
The system further asks the user to input two pieces of information: (1) the date that the customer became a member (in the format DD/MM/YYYY), followed by (2) the points balance value for the newly added customer. The system determines which rank (G/S/B) the customer belongs to by comparing the *current date* (i.e., the date that the program is run) and the date entered by the user. After getting all user inputs, the new customer should be added into the system. For any invalid inputs (e.g., wrong date format/ values, a future date is entered, incorrect range of points values, etc.), the system allows **TWO** more retries. With more than **THREE** times of invalid inputs, the system prints an appropriate error message and returns to the Main Menu.

Delete customer
The system displays the information of the customer (including all the fields), and prompts for user's "Yes/No" confirmation on the delete operation. The customer record is deleted from the system if it is confirmed.

A message showing the summary of the above operation is then displayed, and the system returns to the Main Menu.

~~R4~~ [4] Enter Customer View

[After the starting data is loaded] When the user inputs 4 in the Main Menu, the system prompts for the next user input of a customer ID. If such customer ID does not exist in the system, the system displays an error message and returns to the Main Menu. Otherwise, it

displays the Customer View Menu as shown below, and allows further processing on the particular customer (e.g., DavidChan) as follows (see R4.1 to R4.3 below).

```
Action for Customer ID: DavidChan
***** Customer View Menu *****
[1] Earn CC Points
[2] Redeem Gifts
[3] Modify CC Points Balance
[4] Return to Main Menu
**************************
Option (1 - 4):
```

R4.1 [1] Earn CC Points

When such option is chosen, the system prompts for user input of a floating point value, which is the amount of money spent for converting to CC points. The system then calculates and adds the CC points into the Points Balance for the customer according to the "Points Conversion Rule" (see R7).

R4.2 [2] Redeem Gifts

When such option is chosen, the system displays **all** the gifts under **one of** the Gift Categories according to the user input choice of gift category. Fields about the gift records should be displayed, including the Gift ID, Gift Description, Price, Required Points—the **actual required points for the customer**, which could be a discounted value according to the customer rank, see R7. Gift records are **sorted by the Required Points**, from the smallest to the largest. The display should also identify those gifts that the customer can redeem by only using his available CC points without paying extra money.

The system then allows the user to enter the Gift ID of the gift that he wants to redeem, followed by the amount of CC points used for redemption. For any invalid inputs (e.g., wrong Gift ID, out-of-range CC points for the redemption, etc.), the system allows **TWO** more retries. With more than **THREE** times of invalid inputs, the system prints an appropriate error message and returns to the Customer View Menu.

The user can redeem a gift with "less-than-required" CC points through paying extra money in the redemption transaction (see R7 about the "Points Conversion Rule"). As a result, the user can still redeem a gift if his available CC points is less than the required one for the gift.

After getting all user inputs, the system displays the extra money needed in the redemption transaction and asks for the user's confirmation. Upon user's confirmation, the system subtracts the correct amount of CC points from the Points Balance for the customer.

R4.3 [3] Modify CC Points Balance

When such option is chosen, the system displays the current CC Points Balance of the customer, and asks for user input of a new CC Points Balance value. The system then updates the CC Point Balance for the customer.

R4.4 [4] Return to Main Menu

When such option is chosen, the system returns to the Main Menu.

R4.5 Stay at the Customer View Menu

Following R4.1, R4.2 and R4.3, after an operation on the CC Points Balance of the customer, the system should display the **change** in the CC Points Balance of the customer by that operation, and then stays at the Customer View Menu.

~~R5~~ [5] Show Transaction History

[After the starting data is loaded] When the user inputs 5 in the Main Menu, the system prompts for the next user input of a customer ID. If such customer ID does not exist in the system, the system displays an error message and returns to the Main Menu. Otherwise, it displays all the CC Points transaction records history for that customer since the program starts running:

- All kinds of CC Points transactions (under R4) should be considered and displayed.
- The transaction records should be displayed in the order that they were carried out.
- For an "Add CC Points" transaction, the amount of money spent for earning CC points, and the change in the CC Points Balance, should be displayed.
- For a "Redeem Gifts" transaction, the gift that is redeemed (its gift ID and gift description), the change in the CC Points Balance, and the extra money needed to pay (if any) in the redemption transaction, should be displayed.
- For a "Modify CC Points Balance" transaction, the type (increase or decrease) and change in the CC Points Balance, should be displayed.
- At the end, a summary showing the original CC Points Balance, the final CC Points Balance, the change in CC Points Balance, and the **TOTAL** amount of extra money the customer needs to pay as a result of all redemption transactions should be displayed.

- A meaningful message should be shown if there have been no transactions made so far for the customer.

After displaying the transaction history, the system returns to the Main Menu,

~~R6~~ [6] Credits and Exit

When the user inputs this option, the system prompts for user's confirmation. If the user inputs 'n' or 'N', the system returns to the Main Menu. If the user inputs 'y' or 'Y', the system displays the personal particulars (student name, student ID, tutorial group) of the group members and terminates. Other input is not acceptable and the system should ask the user to confirm again.

R7 Points Conversion Rule

The CC points of a customer could be modified by user operations under R4. The change should be made according to the "Points Conversion Rule" below, which also takes the rank of the customer into consideration:

- A spending of $250 can earn 1 CC point. No CC point can be earned by the remaining spending less than $250.
- Conversion rate during gift redemption: 1 CC point is worth $0.2.
- According to the rank of the customer, there is a discount on the CC points required for a gift in the gift redemption. The final points required is **rounded to nearest integer**:

Rank	Discount
Gold	10% off
Silver	5% off
Bronze	No discount

- The rank of the customer does not affect the price of a gift.
- During the redemption process, if the user redeems a gift using "less-than-required" CC points, the extra money needed is calculated by subtracting the money value of the CC points used in the redemption (calculated using the above conversation rate) from the price of the gift.

  For example, consider a customer who is going to use 1600 CC points to redeem gift D01:

  For a Gold customer:

  Extra money needed = $0

  (1575 CC points is required and to be deducted)

For a Silver or a Bronze customer:
Extra money needed = $350 – $320 = $30
(Silver customer: 1663 CC points is required)
(Bronze customer: 1750 CC points is required)

~~R8~~ Suitable checking on user's input is expected, except in situations with assumptions stated in the requirements above. Appropriate error messages should be printed whenever unexpected situation happens, e.g., input value out of range, incorrect date format, etc.

~~R9~~ The use of functions (in addition to main function) and classes (i.e. OOP design) are expected in your program. Appropriate comments should be added in your source code file.

~~R10~~ Creativity and Critical Thinking: Use suitable format to present all required information of gifts and customers clearly and neatly. Additional features can be added.

~~Tips~~

1. To handle unexpected input error (e.g. input a character to an integer variable), you may use the following code appropriately in your program:

```
cin.ignore();     // Discard the content in the input sequence.
cin.clear();      // Reset the input error status to no error.
```

Video Requirements

This is an individual task under this group project. Each student needs to create a video recording which records either (1) your explanation on the working algorithms of the codes that you designed and wrote, or (2) the testing of project codes (those you wrote or the whole group's work) using test case scenarios. See points below for the specific requirements of the video recording:

- Duration of video is at maximum 2 minutes long. Use **MS Teams** to record.
- The video recording is used to demonstrate your contribution in the group project. If your work done is too much to be all included in the 2-minute video, choose the most important/ representative work to record and explain.
- At the beginning, **clearly mention** the objective of your video: whether you are going to (1) explain the working algorithms of your codes, or (2) run and test your project codes.
- The video should include your voice recording (in English) as you give the presentation (either code explanation or the code testing). Your voice should be clear and loud enough.
- The video should show the computer screen as you give the presentation. Suitable cursor movement or text highlighting by mouse action should be present.

- In addition to your name shown by MS Teams, you should show your English name, Student ID and lecture group on the screen (e.g., put down the information using Notepad and display it). The video should keep showing it for identity verification. It is optional to show your face in the recording.
- Show the whole screen, not just part of the screen. While showing the source codes or code-testing results, make sure the text is clearly shown, large enough and visible.

Submission

Source File: Each group submits one source code file (i.e., **GRS.cpp**).
Video Recording: Each student submits the shared video link via Blackboard, through which your video recording can be viewed by your subject lecturer successfully.
[IMPORTANT: Remember to set up the access right correctly for the shared link. You can refer to the guidelines available in Blackboard on how to prepare the recording and set up the shared link.]
Peer-to-peer Evaluation: Each student fills in the peer-to-peer evaluation form (MS Word) and submits via Blackboard.

All submission should be done **through Blackboard by 23:59, 28 Apr 2024 (Sunday)**. Late submission is subject to 20% deduction in your final marks for each day (including public holidays and Sundays). No late submission is allowed 4 days after the due date.

Components Weighting

1. Program (GRS.cpp)	80%	(Group & Individual)*
2. Video Recording	20%	(Individual)
Total	100%	

* Marks for (1) Program is determined by the group-based marks (80%) and percentage of individual contribution (20%), where the percentage of individual contribution is directly proportion to the average marks given by group members in the peer-to-peer evaluation.

Grading Criteria (Program)

Your program will be executed with different test cases in **Microsoft Visual Studio**. Any deviation from the requirement is considered incorrect and **no mark** is given for that case. Your program will also be marked based on its user-friendliness and creativity (e.g., information display, appropriate prompt messages and action result messages if needed).

Aspects	Percentage
Program correctness (Follow ALL instructions, marks deduction on errors found)	70%
Program design (Appropriate use of functions, use of class, modularity, etc.)	5%
Program standard (Use of variable names, indentation, line spacing, clarity, comments, etc.)	5%
Algorithm design (Use of reasonable algorithms and data structures)	5%
User-friendliness (Clear guidelines to users, messages to users, etc.)	5%
Creativity and critical thinking (Additional useful features)	10%
Total (Group Mark)	**100% (max)**

Note: the length of your program does not affect the grading of the assignment. However, appropriate use of loops and functions are expected to avoid too many repeated codes in your program, which contributes to the program design score of this assignment.

Marks Deduction (Program)

Syntax error: 100% deduction. You will get **0 mark** if your program fails to be compiled.
Runtime error: No mark for the particular test case that triggers the runtime error.
Logic error (bug): No mark for the particular test case that deviates from the requirement. Note that a logic error may lead to **failure in ALL test cases** of one particular set of requirements.

Grading Criteria (Video Recording)

Aspects	Percentage
Basic requirements (video length, language used, showing identity, etc.)	25%
Presentation contents (suitable content, logical presentation, etc.)	50%
Communication clarity (text display, voice, mouse actions, etc.)	25%
Total	**100% (max)**

*** **Ensure the originality of your work. Plagiarism in any form is highly prohibited.** ***

- End -