CS 102 - Object Oriented Programming and Design Methodologies: Assignment #1

Fall Semester 2024

Due on September 14, 2024, 11.59pm

Student Name, ID, Lecture Section

Instructions

1. This homework consists of one large programming exercise.
2. This exercise requires you to study and apply *structures* - a construct that is introduced in Lab 03 of your course.
3. This exercise also requires to think about and explore - how to create and explore elements in an array of structures.
4. Please submit the solution to this Assignment as a single .cpp file (or a zipped archive in case of multiple cpp files), with your name, ID and section clearly marked in the source code.
5. Submission will be via Canvas LMS.
6. No extensions will be given for Assignment Submission.
7. While collaboration and discussion are encouraged, this assignment is strictly individual. Plagiarism and copying, whether partial or complete will result in a *zero* grade for both students, whose assignments are found to be similar. Furthermore, the plagiarism may be reported to the University Conduct Committee as well.

Problem 1

(100 points) [**Bank Simulation**] Write a program in C++ that simulates a **Bank**. For the purpose of this assignment, a bank has the following entities: (1) Bank Account, (2) Customer, (3) Employee

A **Bank Account** has the following features:

1. Account holder's **name**: This is a string with a minimum of 3 and a maximum of 30 characters.
2. Account **number**: It is a unique 8-digit integer (i.e. $10000000 \leq$ account number ≤ 99999999) (To have unique account numbers, consider assigning account numbers incrementally)
3. **Balance** in the account: A double precision number indicating the amount in the account.
4. Account **Status**: This is either '**Active**' or '**Frozen**'.

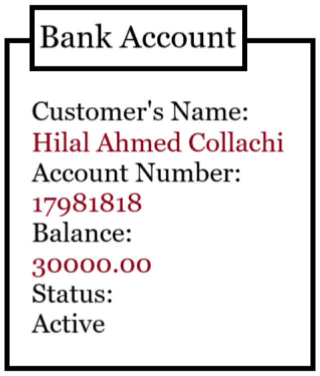
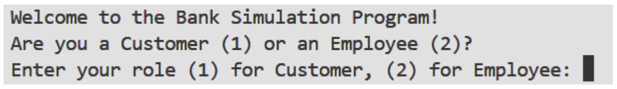


Figure 1: Details of a Bank Account

You must use a `struct` construct to simulate a Bank Account. To operate on the bank account two roles are available: a **Customer** and an **Employee**. Use a menu driven interface to choose the role of a customer or an employee as shown below. (In case of an invalid option, display a message to retry).

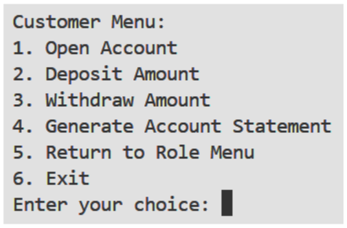


```
Welcome to the Bank Simulation Program!
Are you a Customer (1) or an Employee (2)?
Enter your role (1) for Customer, (2) for Employee: █
```

Figure 2: Console-Based Menu for Selecting Role

If you choose the role of a Customer, the following operations are available (as shown in Figure 3).

1. Open an Account,
2. Deposit Amount (in Account),
3. Withdraw Amount (from Account),
4. Generate Statement (for Account).



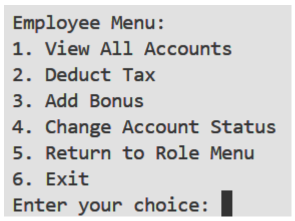
```
Customer Menu:
1. Open Account
2. Deposit Amount
3. Withdraw Amount
4. Generate Account Statement
5. Return to Role Menu
6. Exit
Enter your choice: █
```

Figure 3: Customer Menu

Problem 1 continued on next page. . .

If you choose the role of an Employee, the following operations are available (as shown in Figure 4).

1. View All Accounts,
2. Deduct Tax (from Account),
3. Add Bonus (to Account),
4. Change Account Status.

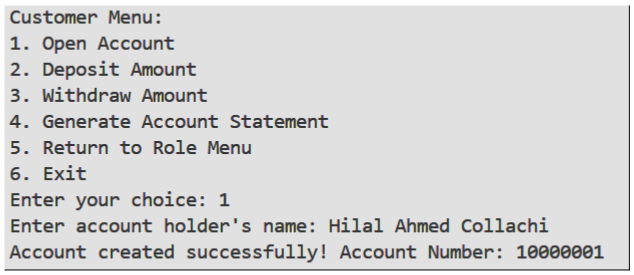


```
Employee Menu:
1. View All Accounts
2. Deduct Tax
3. Add Bonus
4. Change Account Status
5. Return to Role Menu
6. Exit
Enter your choice:
```

Figure 4: Employee Menu

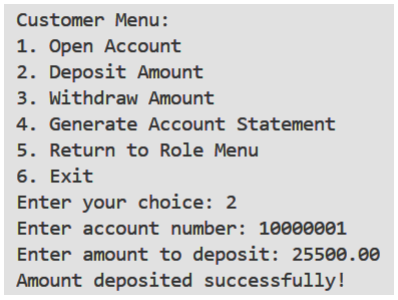
For the role of the **Customer** these options work as follows:

1. **Open an Account**: This option requires the user to enter their name; an account number is generated automatically. The default account status is 'Active'.
2. **Deposit Amount**: This option asks the user to enter an account number and an amount to be deposited into the account. The amount is deposited successfully if the amount ≥ 0.0 and the account status is 'Active'. Screenshots of some of these operations are shown below:



```
Customer Menu:
1. Open Account
2. Deposit Amount
3. Withdraw Amount
4. Generate Account Statement
5. Return to Role Menu
6. Exit
Enter your choice: 1
Enter account holder's name: Hilal Ahmed Collachi
Account created successfully! Account Number: 10000001
```

Figure 5: Create an Account Option



```
Customer Menu:
1. Open Account
2. Deposit Amount
3. Withdraw Amount
4. Generate Account Statement
5. Return to Role Menu
6. Exit
Enter your choice: 2
Enter account number: 10000001
Enter amount to deposit: 25500.00
Amount deposited successfully!
```

Figure 6: Deposit Amount Option

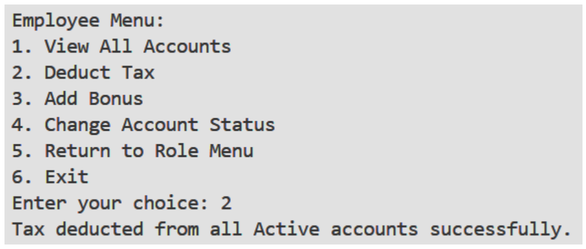
Problem 1 continued on next page. . .

3. **Withdraw Amount**: This option asks the user to enter an account number and an amount to be withdrawn from the account. The amount is withdrawn successfully if the amount $\leq$ the **balance** in the account and the account status is 'Active'.
4. **Generate Statement**: This option asks the user to enter their name. It then generates the details of all accounts with this name - the statement prints: the account holder's name, the account number, the balance in the account and the account status.

For the role of an **Employee** these options work as follows:

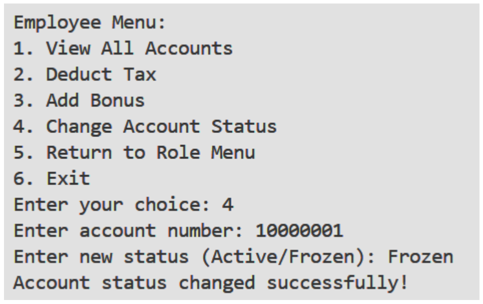
1. **View All Accounts**: This option displays details of all bank accounts in the bank. This includes the name of the account holder, account number, balance in each account and the account status.
2. **Deduct Tax**: This option allows the bank employee to deduct tax from all accounts in the bank that are 'Active'. Note that the tax is 10% of the balance in the account.
3. **Add Bonus**: This option allows the bank employee to add bonus to all accounts in the bank that are 'Active'. Note that the bonus is 10% of the balance in the account.
4. **Change Account Status**: This option allows the employee to change the status of an account from 'Active' to 'Frozen' or from 'Frozen' to 'Active'.

Screenshot of some of these operations is shown as follows:



```
Employee Menu:
1. View All Accounts
2. Deduct Tax
3. Add Bonus
4. Change Account Status
5. Return to Role Menu
6. Exit
Enter your choice: 2
Tax deducted from all Active accounts successfully.
```

Figure 7: Deduct Tax Option



```
Employee Menu:
1. View All Accounts
2. Deduct Tax
3. Add Bonus
4. Change Account Status
5. Return to Role Menu
6. Exit
Enter your choice: 4
Enter account number: 10000001
Enter new status (Active/Frozen): Frozen
Account status changed successfully!
```

Figure 8: Change Account Status Option

Problem 1 continued on next page. . .

Notes

1. You must use a `struct` to create and maintain bank accounts
2. Apart from the use of a structure, you have to create an array of bank accounts to maintain the data of all bank accounts. For reference purposes, you may consult Chapter 4 (Pages 131–162) and Chapter 7 (Page 277–2790) of the book "Object Oriented Programming in C++", 4th Edition by Robert Lafore.
3. It is recommended to use data validation at every step, since incorrect data entry might cause the program to crash resulting in loss of data. Examples are: if an incorrect option is chosen, an appropriate message is displayed and the user is asked to re-enter the data.
4. Use proper naming convention for all variable and function names. Either use snake_case or use CamelCase for names. Likewise, you may name your file CS224HW1.cpp for uniformity. Make sure to put appropriate comments identifying various functions and variables. Finally, include your name, ID and Section as a comment at the top of your file.

Points Distribution

The following are the points distribution for each component of the assignment:

Component	**Points**
Bank Account Structure	25
Array (or Collection) of Bank Accounts	10
Customer Operations	25
Employee Operations	25
Customer Menu	5
Employee Menu	5
Role Menu	5
TOTAL	100