Comp Sci 371: Advanced Object-oriented Design
Group Project
Full Mark: 100
Code Submission Deadline: Day of your presentation

Advanced Bank Management System

Objective: To develop a C++ object-oriented program to implement a bank management system.

Project Overview:

> Your task is to develop a Bank Management System in C++ that encompasses user accounts, manager accounts, transactions, and a login system. The system involves three main classes: User, Transaction, and BankAccount. Additionally, there is a special class for bank managers.
>
> The following are the main requirements of this project. **You are allowed to implement as many classes as you want to fulfill the following requirements**.

Requirements:

- Implement a class to manage user bank accounts, including methods to create and delete users, load users' information from a file, and handle user logins. A typical bank account should have attributes such as account number, account type, balance, and customer information. This class will include a static data member to keep track of the number of active accounts currently held by the bank.

- User login information and account information should be stored in the users.txt file.

- Create a class to manage banking accounts, allowing users to perform actions like withdrawal, deposit, and printing account summaries.

    - The Withdraw function verifies the availability of funds and deducts a specified amount from the user's account. If there are insufficient funds, the user is promptly notified with an error message.

- Deposit enables the user to inject a specified amount of money into their account.

- PrintAccountSummary exhibits the user's comprehensive transaction history along with their present account balance.

- Develop a specialized class for bank managers, providing login functionality and access to all user data.

- The login data for managers is stored in managers.txt. No class will have the capability to create a manager account; instead, the manager's name and password must be manually inserted into the managers.txt file. Initially, the first names of you and your group members, along with their corresponding passwords, will be stored inside managers.txt.

Your program should start with the following options:

1. User Login

2. Create Account

3. Manager Login

4. Exit

For any further clarification, please contact your instructor.

Your code and its functionality will be evaluated according to the following criteria

✓	Use of an Abstract Class	5 marks
✓	Use of classes, objects, and constructors (default, parametric, copy)	10 marks
✓	Use of a static member in the account class	5 marks
✓	Use of Encapsulation, Polymorphism, and Inheritance	10 marks
✓	Use of pointers	5 marks
✓	File, and memory management	5 marks
✓	Correctly working program	10 marks

Rubrics

Your code and its correct functionalities	50
Report	20
Presentation	25
Peer Review	5

Functionalities

You need to implement the topics, concepts, and syntax that we covered in class. Examples include pointers, class definitions, header files, encapsulation, polymorphism, inheritance, File I/O, and strings.

Report

You are required to submit a comprehensive report detailing your project implementation, methodology, inputs/outputs, and any insights gained throughout the development process. The report should be structured and formatted in a professional manner. You can follow the guidelines provided below:

1. Introduction
2. Methodology
    - Describe the approach you took to design and implement your project.
    - Explain the data structures, and programming techniques used.
3. Implementation Details:
    - Provide a detailed explanation of how you implemented the project.
    - You can use UML, detailed class design, attributes, and class behaviors here to describe your project.
    - Discuss any challenges encountered and how you addressed them.
    - Include code snippets or pseudocode to illustrate key components of your implementation.
4. Input/Output
    - Provides inputs and outputs of your project which help to test all features of your project.
5. Conclusion:
    - Summarize the project, key findings, and insights gained from your project.
6. References:
7. Appendices (Code):
    - Add the entire code of your project.

Presentation

In Week 14, you will present your project during the class hour. Each group will have approximately 15-20 minutes for their presentation. This will consist of 12-15 minutes for the presentation itself and 5 minutes for answering questions. Your presentation slides should include information on how you fulfill the project requirements, such as the utilization of pointers or inheritance in your project, along with the corresponding code implementation. You should clearly indicate which group member wrote which specific parts of the code in your program. During the presentation, you'll be asked to run your code and conduct a code walk-through, with each member presenting the sections they implemented. You should use your laptop to present it, and make sure your laptop is compatible with the class projector.

Peer Review Report

In your peer review report, describe the challenges you faced in completing the project and the approaches you used to address them. Also include comments on your group members' participation and cooperation.

Each student will evaluate their group members using the criteria below (scale of 1–4).

1) Effort in researching and developing the project (1–4)

1. The student did not collect any information related to the research topic, according to the rubric provided earlier.
2. The student collected very little information related to the research topic, according to the rubric provided earlier.
3. The student collected a reasonable amount of information, and most of it was relevant to the research topic, according to the rubric provided earlier.
4. The student collected a great deal of information, and all of it was relevant to the research topic, according to the rubric provided earlier.

2) Sharing information and code with the group (1–4)

1. The student did not share any information with teammates.
2. The student shared very little information related to the research topic.
3. The student shared some information, and most of it was related to the research topic.
4. The student shared a great deal of information, and all of it was related to the research topic.

3) Performance of assigned duties (1–4)

1. Did not perform any duties of the assigned team role.
2. Performed very few duties.
3. Performed nearly all duties.
4. Performed all assigned duties.

4) Sharing the workload (1–4)

1. Always relied on others to do the work.
2. Rarely completed assigned work and often needed reminders.
3. Usually completed assigned work and rarely needed reminders.
4. Always completed assigned work without needing reminders.

5) Attendance and communication (1–4)

1. Missed most group meetings and did not respond to communications.
2. Frequently missed group meetings and seldom responded to team communications.
3. Attended most meetings and responded to communications within a couple of days.
4. Attended all group meetings and responded to communications in a timely manner.

In addition to these ratings, please include any comments or feedback you have about the group's work and individual contributions.