University of Lincoln
School of Computer Science
CMP9133M – Advanced Programming
Workshop 8

Task (assessed)

You are tasked with creating a banking system using C++ classes. Your goal is to demonstrate encapsulation, access modifiers, and the relationship between different classes. Create classes for accounts, customers, and the banking system, and implement methods to perform various banking operations while enforcing encapsulation.

Instructions:

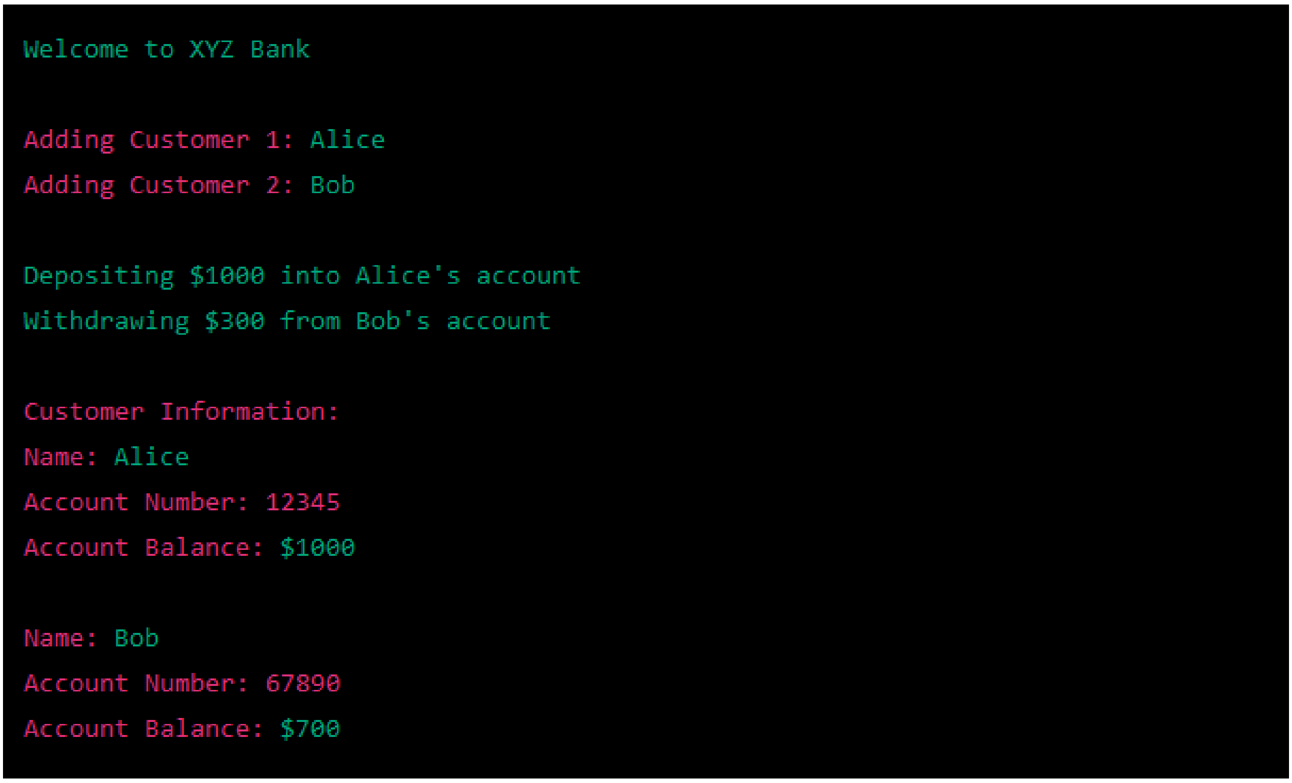
1. Create a class named `Account` with private attributes:
   - `int accountNumber`: to store the account number.
   - `double balance`: to store the account balance. Implement public member functions:
   - `Account(int accountNumber)`: Constructor to initialize the account details.
   - `int getAccountNumber() const`: Getter method for the account number.
   - `double getBalance() const`: Getter method for the account balance.
   - `void deposit(double amount)`: Method to deposit money into the account.
   - `void withdraw(double amount)`: Method to withdraw money from the account.
2. Create a class named `Customer` with private attributes:
   - `std::string name`: to store the customer's name.
   - `Account account`: to store the customer's associated account. Implement public member functions:
   - `Customer(const std::string &name, int accountNumber)`: Constructor to initialize the customer details and create an associated account.
   - `std::string getName() const`: Getter method for the customer's name.
   - `int getAccountNumber() const`: Getter method to get the associated account number.
   - `double getAccountBalance() const`: Getter method to get the associated account balance.
   - `void deposit(double amount)`: Method to deposit money into the associated account.
   - `void withdraw(double amount)`: Method to withdraw money from the associated account.
3. Create a class named `Bank` with private attributes:

    - `std::string bankName`: to store the bank's name. Implement public member functions:
    - `Bank(const std::string &name)`: Constructor to initialize the bank's name.
    - `void addCustomer(const std::string &name, int accountNumber)`: Method to add a new customer to the bank.
    - `void displayCustomerInfo(int accountNumber) const`: Method to display the information of a customer by account number.

4. In the `main()` function:
    - Create an instance of the `Bank` class.
    - Add customers to the bank and perform various banking operations.
    - Display customer information and account balances

Note: Ensure that the private attributes of the classes are encapsulated and accessed using the appropriate getter methods. Use appropriate access modifiers to enforce encapsulation. Demonstrate the relationship between customers and their associated accounts. You can use the skeletonised script `banking_system.cpp` that you find on blackboard to solve this task.

Test case:



```
Welcome to XYZ Bank

Adding Customer 1: Alice
Adding Customer 2: Bob

Depositing $1000 into Alice's account
Withdrawing $300 from Bob's account

Customer Information:
Name: Alice
Account Number: 12345
Account Balance: $1000

Name: Bob
Account Number: 67890
Account Balance: $700
```