upGrad ENTERPRISE

Assignment Case Study: Bank Account Management

Problem Statement: Bank Account Management

Objective: Design and implement a C# program to manage bank accounts for customers, allowing operations such as depositing and withdrawing money, while ensuring proper data encapsulation and validation.

Description:

A bank requires a system to handle customer accounts. Each account should store details such as the account number, account holder's name, and current balance. The system should allow users to perform basic banking operations like depositing and withdrawing money, with appropriate checks to ensure valid transactions. The program should demonstrate the use of Object-Oriented Programming (OOP) concepts, specifically Class, Object, Constructor, Properties, and the this keyword.

Requirements:

1. **Class**: Create a `BankAccount` class to represent a bank account.

2. *Properties*:

- `AccountNumber`: A unique identifier for the account (read-only outside the class).

- `AccountHolder`: The name of the account holder (read-write).

- `Balance`: The current balance of the account (read-only outside the class, modifiable only through specific methods).

3. *Constructor*: Initialize an account with an account number, account holder's name, and an initial balance. Ensure the initial balance is non-negative.

4. *Methods*:

- `Deposit`: Allow depositing a positive amount into the account, updating the balance.

- `Withdraw`: Allow withdrawing an amount from the account, provided it is positive and does not exceed the current balance.

- `DisplayAccountInfo`: Display the account details, including account number, account holder's name, and current balance.

5. *this Keyword*: Use the `this` keyword to differentiate between instance properties and constructor/method parameters where applicable.

6. *Validation*:

- Ensure deposits and withdrawals are positive amounts.

- Prevent withdrawals that exceed the current balance.

- Initialize the balance to zero if a negative initial balance is provided.

7. *Main Program*: Create an instance of the `BankAccount` class and demonstrate its functionality by performing deposits, withdrawals, and displaying account details.

Expected Output:

The program should create a bank account object and allow the following interactions:

- Display initial account details.

- Perform a valid deposit and show the updated balance.

- Perform a valid withdrawal and show the updated balance.

- Attempt an invalid withdrawal (e.g., exceeding the balance) and display an error message.

- Display final account details.

Constraints:

- The account number should be a string and immutable after creation.

- The balance should be a decimal value to handle monetary amounts accurately.

- All monetary transactions (deposit/withdrawal) must be positive and properly validated.

- Use C# properties to encapsulate data, restricting direct modification where appropriate.

Example Interaction:

- Create an account with account number "12345", account holder "Narasimha", and initial balance 1000.00.

- Deposit 500.00.

- Withdraw 200.00.

- Attempt to withdraw 2000.00 (should fail).

- Display the account details after each operation.