Answer to Study Prompts

- **Run your Console application, what is it doing? What is being outputted to the console?**

This Console application is interacting with the bank accounts of a customer.

There are two bank accounts - DodgyBankAccount and SecureBankAccount.

First, it goes to DodgyBankAccount.
It creates an instance with an account number that is a constant one.
The methods/work it does includes,

- Getting the account balance
- Depositing the amount and viewing the account balance with Rewards if any.
- Adding the rewards without any deposit and viewing the account balance after reward points.
- Finally, Trying to directly modify the account balance and got succeeded.

Next, to SecureBankAccount.

In this,

- It is getting the account balance.
- Depositing the amount and viewing the account balance with Rewards if any.
- Trying to add the rewards without a deposit but not able to do it.
- Displaying the account balance again
- Trying to directly modify the account balance. Since it is a secure bank it doesn't work here.

- **Look at the DodgyBankAccount, this class is not well-encapsulated. Can you note down the problems with how the class is designed, and the ways it is being misused?**

DodgyBankAccount is not encapsulated at all.
Every attribute and method is accessible from the outside.

The problems are

- The account number is modifiable.
- The account balance can be changed directly without any deposit amount
- The Reward amount can be modified to our wish.

- **Compare and contrast the DodgyBankAccount and the SecureBankAccount, how is the SecureBankAccount different to the DodgyBankAccount? How is it designed to prevent it from**

being misused? Are there instances of better method names for clearer abstraction?

SecureBankAccount is well encapsulated.

- Here the attributes are declared private.
- The account number is declared as a read-only string so it cant be modified.
- Account balance and reward amount are also private so unlike DodgyBankAccount they cant be changed.
- Add Reward method also has the private access modifier, so we cant add the reward amount without depositing.
- Also have a better method names from GetAccountBalanceDetails to DisplayAccountBalanceDetails.