Encapsulation – Study Prompts

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

    - The output shows the same set of actions performed to two different bank accounts and getting different outcomes. The first one appears to be insecure while the second one seems more secure.

- 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?

    - The properties `AccountNumber`, `AccountBalance` and `RewardAmount` are public, but these should be private. If `RewardAmount` is fixed, it should be made const.

    - The `AddReward`() method should be private since it should not be accessible through an object but only while making a deposit.

- 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?

`DodgyBankAccount`	`SecureBankAccount`
Properties are public	Properties are private
Account number can be changed	Account number is readonly
RewardAmount can be changed	RewardAmount is constant
AddReward() is public which makes it expose a feature that should only happen when making a deposit	AddReward() is private
GetAccountBalanceDetails() could be named better since it does not return anything	DisplayAccountBalanceDetails() named appropriately since it outputs the balance to the console – clearer abstraction