

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

In this case, the console application is dealing with the customer with two bank accounts (one is secure bank account and the other one is dodgy bank account).

It takes the account number("ABCDEF123") and goes to DodgyBankAccount and do the operations there like:

- 1) Getting the account balance.
- 2) Depositing the amount and getting the reward points.
- 3) Adding the rewards without depositing the amount.
- 4) Changing the bank balance and directly.

Coming to SecureBankAccount :

- 1) Checking for the current balance.
- 2) Depositing the amount and getting the reward points.
- 3) Trying to add the rewards points but cannot be able to do it.
- 4) Showing the bank balance.
- 5) Trying to modify the bank balance but cannot be able to do it.

## **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 looks like not well-encapsulated as the members of the class are easily accessible from the outside of class. Nothing is hidden here.

The problems with how the class is designed, and the ways it is being misused:

- 1) The account number can be modified easily.
- 2) The deposit amount and the rewards can be changeable.
- 3) Can easily change the rewards even if they don't deposit also.

## **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?**

In SecureBankAccount the attributes are declared as private hence can say that it is well-encapsulated. The account number here is readonly so it's not easy to change. Also, the reward amount is constant, so its hard to modify. There is a private access modifier for the AddReward() method, because of this we cant modify/add the rewards to the account without a proper deposit. Also in the DodgyBankAccount, we have GetAccountBalanceDetails() name for the method and in SecureBankAccount, we have DisplayAccountBalanceDetails() name for the method. The DisplayAccountBalanceDetails() sounds much better than GetAccountBalanceDetails() as get is usually referred to return a value and display for printing something.

