Part B: Small-Class Design

1. Class Skeleton

Outline a small class C++ that represents a BankAccount.

Include:

Two private data members (e.g., balance, accountNumber).

One or two public methods that allow interaction with the balance (e.g., deposit or withdraw).

```
Class BankAccount {

Int balance;

Int accountNumber;

Public:

Int deposit;

};
```

2. Encapsulation Justification

For each private data member, explain in 1–2 sentences why it should be kept private.

Balance should be kept private because you do not want the number changed without going through the proper steps. The balance should not be changed unless there is a deposit. AccountNumber should be kept private because it does not change. The accountNumber will always be the same so encapsulation makes sure it can't be altered.

For each public method, explain how it enforces constraints or validations before modifying any private data.

Deposit must be called before it can modify balance.

3. Documentation

Briefly show how you would document the class or methods so other developers understand they must not directly manipulate the balance.

I would put a comment next to balance that says //only withdrawal and deposit should affect balance.