Encapsulation:

```
class BankAccount:
    def __init__(self, account_number, name, balance):
        self.__account_number = account_number
        self.__name = name
        self.__balance = balance
    def get_account_number(self):
        return self.__account_number
    def get_name(self):
        return self.__name
    def set_name(self, new_name):
        self.__name = new_name
    def get_balance(self):
        return self.__balance
    def deposit(self, amount):
        if amount > 0:
            self.__balance += amount
            print(f"Deposited ₹{amount}. New Balance: ₹{self.__balance}")
        else:
            print("Invalid deposit amount")
    def withdraw(self, amount):
        if 0 < amount <= self.__balance:
            self.__balance -= amount
            print(f"Withdrawn ₹{amount}. New Balance: ₹{self.__balance}")
        else:
            print("Insufficient balance or invalid amount")
    def show_details(self):
        print("=== Account Details ===")
        print(f"Account No: {self.__account_number}")
        print(f"Name: {self.__name}")
        print(f"Balance: ₹{self.__balance}")
acc1 = BankAccount(12345, "Rahul", 5000)
acc1.show_details()
print("Account Number (via getter):", acc1.get_account_number())
print("Name (via getter):", acc1.get_name())
print("Balance (via getter):", acc1.get_balance())
acc1.deposit(1500)
acc1.withdraw(2000)
acc1.set_name("Rahul Kumar")
acc1.show_details()
```

output:

```
=== Account Details ===
Account No: 12345
Name: Rahul
Balance: ₹5000
Account Number (via getter): 12345
Name (via getter): Rahul
Balance (via getter): 5000
Deposited ₹1500. New Balance: ₹6500
Withdrawn ₹2000. New Balance: ₹4500
=== Account Details ===
Account No: 12345
```

Name: Rahul Kumar

Balance: ₹4500