Case Study Title: Banking System Application Using OOPs Concepts

1.BankOperations (Interface)

```
package Day2Task;


public interface BankOperations {
    void deposit(double amount);
    void withdraw(double amount);
    void checkBalance();
}
```

2. Account (Abstract Class)

```
package Day2Task;


public abstract class Account implements BankOperations {
    String accountNumber;
    double balance;

    public Account(String accNo, double balance) {
        this.accountNumber = accNo;
        this.balance = balance;
    }

    public void checkBalance() {
        System.out.println("Current Balance in " + accountNumber + ": ₹" + balance);
```

```
    }

    public void transfer(Account target, double amount) {
        if (balance >= amount) {
            this.withdraw(amount);
            target.deposit(amount);
            System.out.println("Transferred ₹" + amount + " from " + this.accountNumber + " to " +
target.accountNumber);
        } else {
            System.out.println("Insufficient Balance to Transfer");
        }
    }
}
```

3.SavingsAccount (extends Account, implements BankOperations)

```
package Day2Task;

public class SavingsAccount extends Account {
    final double MIN_BAL = 1000.0;

    public SavingsAccount(String accNo, double balance) {
        super(accNo, balance);
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited ₹" + amount + " to Savings Account");
    }
```

```
    public void withdraw(double amount) {
        if (balance - amount >= MIN_BAL) {
            balance -= amount;
            System.out.println("Withdrawn ₹" + amount + " from Savings Account");
        } else {
            System.out.println("Minimum balance ₹1000.0 required!");
        }
    }
}
```

4.CurrentAccount (extends Account, implements BankOperations)

```
package Day2Task;

public class CurrentAccount extends Account {
    final double OVERDRAFT = 2000.0;

    public CurrentAccount(String accNo, double balance) {
        super(accNo, balance);
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited ₹" + amount + " to Current Account");
    }

    public void withdraw(double amount) {
        if (balance - amount >= -OVERDRAFT) {
            balance -= amount;
```

```
            System.out.println("Withdrawn ₹" + amount + " from Current Account (may use overdraft)");
        } else {
            System.out.println("Overdraft limit ₹2000.0 exceeded!");
        }
    }
}
```

5.Customer

```
package Day2Task;

public class Customer {
    String name;
    SavingsAccount savings;
    CurrentAccount current;

    public Customer(String name, SavingsAccount s, CurrentAccount c) {
        this.name = name;
        this.savings = s;
        this.current = c;
    }

    public void showAccounts() {
        savings.checkBalance();
        current.checkBalance();
    }
}
```

6.BankBranch

```
package Day2Task;


public class Main {
    public static void main(String[] args) {
        // Create Accounts
        SavingsAccount sAcc = new SavingsAccount("S001", 5000);
        CurrentAccount cAcc = new CurrentAccount("C001", 2000);

        // Create Customer
        Customer c1 = new Customer("Alice", sAcc, cAcc);
        System.out.println("✅ Customer Created: " + c1.name);

        // Operations
        sAcc.deposit(2000);
        cAcc.withdraw(2500);
        sAcc.transfer(cAcc, 1000);

        // Final Balances
        c1.showAccounts();
    }
}
```