8] Bank Account

```
import java.util.Scanner;

class Account
{
    String customerName;
    long accountNumber;
    String accountType;
    double balance;

    public Account(String customerName, long accountNumber,
    String accountType, double balance)
    {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

    public void deposit(double amount)
    {
        balance += amount;
        SOP("Deposit successful");
        SOP("Updated balance:" + balance);
    }

    public void displayBalance()
    {
        SOP("Account Number:" + accountNumber);
        SOP("Customer Name:" + customerName);
        SOP("Account Type:" + accountType);
        SOP("Balance:" + balance);
    }
}

class SavAct extends Account
{
    public SavAct(String customerName, long accountNumber,
    double balance)
    {
        SOP(customerName, accountNumber, "Savings", balance);
    }
```

```
public void computeAndDepositInterest(double rate)
{
    double interest = balance * rate /100;
    balance += interest;
    SOP("Interest computed and deposited");
    SOP("Updated balance:" + balance);
}
public void withdraw(double amount)
{
    if(amount <= balance)
    {
        balance -= amount;
        SOP("Withdrawal successful");
        SOP("Updated balance:" + balance);
    }
    else
    {
        SOP("Insufficient funds.");
        SOP("Withdrawal failed");
    }
}
}

public class CurrAcct extends Account
{
    double minimumBalance;
    double serviceCharge;
    public CurrAcct(String customerName, long accountNumber,
    double balance, double minimumBalance, double serviceCharge)
    {
        super(customerName, accountNumber, "Current", balance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }
    private void checkMinimumBalance()
{
    if(amount <= balance)
    {
        balance -= amount;
        if(balance < minimumBalance)
        { balance -= serviceCharge;
```

```
        SOP("Minimum balance not maintained.")
        SOP("Service charge imposed")
        SOP("Updated balance:" + balance);
    }
}
public void withdraw(double amount)
{   if(amount <= balance)
    {   balance -= amount;
        SOP("Withdrawal successful.")
        SOP("Updated balance:" + balance);
        checkMinimumBalance();
    }
    else
    {   SOP("Insufficient funds. Withdrawal failed");
    }
}

public class Bank
{
    psvm(String[] args)
    {   Scanner S1 = new Scanner(System.in);
        SOP("Enter customer name for savings Account:");
        String SCN = S1.nextLine();
        System.out.println("Enter account number for
        savings account");
        long SAN = S1.nextLong();
        SOP("Enter initial balance for savings Account:");
        double SIB = S1.nextDouble();
        SAVAcct SA = new SavAcct(SCN, SAN, SIB);
        SOP("Enter customer name for Current Account:");
        String CCN = S1.next();
        SOP("Enter account number for Current Account:");
        long CAN = S1.nextLong();
        SOP("Enter initial balance for current Account");
```

```
double CIB = S1.nextDouble();
SOP("Enter minimum balance for current Account : ");
double SC = S1.nextDouble();
CurrAcct CA = new CurrAcct (CCN, CAN, CIB, MB, SC);

SOP("Enter deposit amount for savings Account : ");
double SDA = S1.nextDouble();
SA.deposit (SDA);

SOP("Enter interest rate for savings Account : ");
double SIR = S1.nextDouble();
SA.computeAndDepositInterest (SIR);

SOP("Enter deposit amount for current Account : ");
double CDA = S1.nextDouble();
CA.deposit (CDA);

SOP("Enter withdrawal amount for current Account : ");
double CWA = S1.nextDouble();
CA.withdrawal (CWA);

SOP("Final Balances :");
SOP("Savings Account :");
SA.displayBalance();

SOP("Current Account :");
CA.displayBalance();
}
}
```

Output:

Enter customer name for Savings Account : Ravi
Enter account number for Savings Account : 202015
Enter initial balance for savings Account : 10000

Enter customer name for current Account : Reetu
Enter account number for current Account : 202016
Enter initial balance for current Account : 1000
enter minimum balance for current Account : 5000
Deposit Successful.
Updated balance : 15000

Enter interest rate for saving Account : 4
Interest computed and deposited.
Updated balance : 15600.0

Withdraw
Enter withdrawal amount for savings Account : 700
Withdrawal Successfull-
Updated balance : 14900.0

22/01/24

BANK

```
import java.util.Scanner;

class Account {
    protected String name;
    protected int accno;
    protected double balance;

    public void get_info() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Name: ");
        name = sc.nextLine();
        System.out.print("Enter Account Number: ");
        accno = sc.nextInt();
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Amount deposited successfully.");
    }

    public void display() {
        System.out.println("Name: " + name);
        System.out.println("Account Number: " + accno);
        System.out.println("Balance: " + balance);
    }
}

class Cur_acct extends Account {
    private final double min_balance = 500;
    private final double penalty = 100;


    public void withdraw(double amount) {
        if (balance - amount >= min_balance) {
            balance -= amount;
            System.out.println("Amount withdrawn successfully.");
        } else {
            System.out.println("Insufficient balance for
withdrawal.");
        }
        check_min_balance();
    }

    private void check_min_balance() {
        if (balance < min_balance) {
```

```
            balance -= penalty;
            System.out.println("Penalty imposed for falling below
minimum balance.");
        }
    }
}

class Sav_acct extends Account {
    private final double interest_rate = 0.04;

    public void compute_interest() {
        double interest = balance * interest_rate;
        balance += interest;
        System.out.println("Interest credited successfully.");
    }
}

class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter 1 for Current Account or 2 for
Savings Account: ");
        int choice = sc.nextInt();

        Account acc;
        if (choice == 1) {
            acc = new Cur_acct();
        } else {
            acc = new Sav_acct();
        }

        acc.get_info();

        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display Balance");
            System.out.println("4. Compute Interest (Savings Account
only)");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            int choice2 = sc.nextInt();

            switch (choice2) {
```

```
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double amount = sc.nextDouble();
                    acc.deposit(amount);
                    break;
                case 2:
                    if (acc instanceof Sav_acct) {
                        System.out.println("Withdrawal not allowed
for Savings Account.");
                    } else {
                        System.out.print("Enter amount to withdraw:
");
                        amount = sc.nextDouble();
                        ((Cur_acct) acc).withdraw(amount);
                    }
                    break;
                case 3:
                    acc.display();
                    break;
                case 4:
                    if (acc instanceof Sav_acct) {
                        ((Sav_acct) acc).compute_interest();
                    } else {
                        System.out.println("Interest computation not
applicable for Current Account.");
                    }
                    break;
                case 5:
                    System.exit(0);
                default:
                    System.out.println("Invalid choice.");
            }
        }
    }
}
```

OUTPUT :

```
Enter 1 for Current Account or 2 for Savings Account:
1
Enter Name: Clara
Enter Account Number: 1122334455

Menu:
```

BANK

```
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 1
Enter amount to deposit: 1000
Amount deposited successfully.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 2
Enter amount to withdraw: 500
Amount withdrawn successfully.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Name: Clara
Account Number: 1122334455
Balance: 500.0

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 5
PS C:\Users\ADMIN\Documents\CSE III\java prgms> cd
"c:\Users\ADMIN\Documents\CSE III\java prgms\" ; if ($?) { javac
Bank.java } ; if ($?) { java Bank }
Enter 1 for Current Account or 2 for Savings Account:
2
Enter Name: Rosy
Enter Account Number: 101202303

Menu:
```

BANK

```
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 1
Enter amount to deposit: 5000
Amount deposited successfully.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 2
Withdrawal not allowed for Savings Account.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 500
Invalid choice.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 4
Interest credited successfully.

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 3
Name: Rosy
Account Number: 101202303
```

BANK

```
Balance: 5200.0

Menu:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account only)
5. Exit
Enter your choice: 5
```