Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;


class Account {
    String customerName;
    int accountNumber;
    String typeOfAccount;
    double balance;

    Account(String customerName, int accountNumber, String typeOfAccount) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.typeOfAccount = typeOfAccount;
    }

    void details() {
        System.out.println();
        System.out.println("Name: " + customerName);
        System.out.println("Acc no.: " + accountNumber);
```

```
        System.out.println("Type: " + typeOfAccount);
        System.out.println("Balance: " + balance);
    }
}

class SavAcc extends Account {
    Scanner sc = new Scanner(System.in);

    SavAcc(String customerName, int accountNumber, String typeOfAccount) {
        super(customerName, accountNumber, typeOfAccount);
    }

    void details() {
        super.details();
        System.out.println("Minimum Balance: No Minimum Balance for Savings Account");
    }

    void acceptDeposit() {
        System.out.println();
        System.out.println("Enter amount to be deposited");
        double deposit = sc.nextDouble();
        balance += deposit;
        System.out.println("\nTransaction Successfull!!\n");
        System.out.println("Updated Balance: " + balance);
    }

    void permitWithdrawal() {
        System.out.println();
        System.out.println("Withdrawal amount");
        double withdraw = sc.nextDouble();
```

```
        if (balance == 0) {
            System.out.println("\nTransaction Failed");
            System.out.println("Zero Balance");
            return;
        }
        balance -= withdraw;
        System.out.println("\nTransaction Successfull!!\n");
        System.out.println("Updated Balance: " + balance);
    }

    void interest() {
        System.out.println("Months");
        double month = sc.nextInt();
        month /= 3;
        balance = balance + (balance * 0.10 * (month / 4));
        System.out.println("Updated Balance after depositing interest: " + balance);
    }
}

class CurAcc extends Account {
    double minBalance = 2000;
    Scanner sc = new Scanner(System.in);
    CurAcc(String customerName, int accountNumber, String typeOfAccount) {
        super(customerName, accountNumber, typeOfAccount);
    }

    void details() {
        super.details();
        System.out.println("Minimum Balance: " + minBalance);
    }
```

```
void acceptDeposit() {
    System.out.println();
    System.out.println("Enter amount to be deposited");
    double deposit = sc.nextDouble();
    balance += deposit;
    System.out.println("\nTransaction Successfull!!\n");
    System.out.println("Updated Balance: " + balance);
}

void permitWithdrawal() {
    System.out.println();
    System.out.println("Withdrawal amount");
    double withdraw = sc.nextDouble();
    if (balance == 0) {
        System.out.println("\nTransaction Failed");
        System.out.println("Zero Balance");
        return;
    }
    balance -= withdraw;
    if (balance < minBalance) {
        System.out.println();
        System.out.println("oppps!!  balance is less than minimum balance");
        System.out.println("You have to pay penalty of Rs " + 1000);
        balance -= 1000;
        System.out.println("Updated Balance after deducting penalty: " + balance);
        return;
    }
    System.out.println("\nTransaction Successfull!!\n");
    System.out.println("Updated Balance: " + balance);
```

```
    }
}

class Program5 {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int type;

        System.out.print("Enter account holder name: ");
        String name = sc.nextLine();

        System.out.print("Enter SavingsAccount number: ");
        int AccNo1 = sc.nextInt();
        SavAcc sav = new SavAcc(name, AccNo1, "SavingsAccount");// Creating
SavingsAccount object

        System.out.print("Enter CurrentAccount number: ");
        int AccNo2 = sc.nextInt();
        CurAcc curr = new CurAcc(name, AccNo2, "CurrentAccount");// Creating
CuurentAccount object

        // Menu
        while (true) {
          System.out.println("\n1.AccountDetails\n2.Deposit\n3.WithDraw\n4.Interest\n5.Exit");
            System.out.print("Enter your Choice: ");
            int ch = sc.nextInt();
            switch (ch) {
                case 1:
                    System.out.println("\nAccount Type");
                    System.out.println("1.Savings Acc \n2.Current Acc");
                    type = sc.nextInt();
```

```
    if (type == 1) {
        sav.details();
    } else if (type == 2) {
        curr.details();
    }
    break;
case 2:
    System.out.println("\nAccount Type");
    System.out.println("1.Savings Acc \n2.Current Acc");
    type = sc.nextInt();
    if (type == 1) {
        sav.acceptDeposit();
    } else if (type == 2) {
        curr.acceptDeposit();
    }
    break;
case 3:
    System.out.println("\nAccount Type");
    System.out.println("1.Savings Acc \n2.Current Acc");
    type = sc.nextInt();
    if (type == 1) {
        sav.permitWithdrawal();
    } else if (type == 2) {
        curr.permitWithdrawal();
    }
    break;
case 4:
    System.out.println("\nAccount Type");
    System.out.println("1.Savings Acc \n2.Current Acc");
    type = sc.nextInt();
```

```
                if (type == 1) {
                    sav.interest();
                } else if (type == 2) {
                    System.out.println("\nSorry CurrentAccount don't have interst facility");
                }
                break;
            case 5:
                System.exit(0);
                break;
            default:
                System.out.println("Invalid choice");
        }
    }
  }
}
```

OUTPUT:

Command Prompt

```
C:\Users\Admin\Desktop\VIGNESH-1BM21CS240\JAVA>javac Program5_Bank.java

C:\Users\Admin\Desktop\VIGNESH-1BM21CS240\JAVA>java Program5

Enter account holder name: VIGNESH
Enter SavingsAccount number: 123
Enter CurrentAccount number: 456

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 2

Account Type
1.Savings Acc
2.Current Acc
1

Enter amount to be deposited
50000

Transaction Successfull!!

Updated Balance: 50000.0

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 2

Account Type
1.Savings Acc
2.Current Acc
2

Enter amount to be deposited
100000

Transaction Successfull!!

Updated Balance: 100000.0

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 3

Account Type
1.Savings Acc
2.Current Acc
1

Withdrawal amount
25000
```

Command Prompt

```
Transaction Successfull!!

Updated Balance: 25000.0

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 3

Account Type
1.Savings Acc
2.Current Acc
2

Withdrawal amount
10000

Transaction Successfull!!

Updated Balance: 90000.0

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 3

Account Type
1.Savings Acc
2.Current Acc
2

Withdrawal amount
90000

oppps!!  balance is less than minimum balance
You have to pay penalty of Rs 1000
Updated Balance after deducting penalty: -1000.0

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 4

Account Type
1.Savings Acc
2.Current Acc
1
Months
5
Updated Balance after depositing interest: 26041.666666666668
```

```
1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 1

Account Type
1.Savings Acc
2.Current Acc
1

Name: VIGNESH
Acc no.: 123
Type: SavingsAccount
Balance: 26041.666666666668
Minimum Balance: No Minimum Balance for Savings Account

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 1

Account Type
1.Savings Acc
2.Current Acc
2

Name: VIGNESH
Acc no.: 456
Type: CurrentAccount
Balance: -1000.0
Minimum Balance: 2000.0

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 5

C:\Users\Admin\Desktop\VIGNESH-1BM21CS240\JAVA>
```