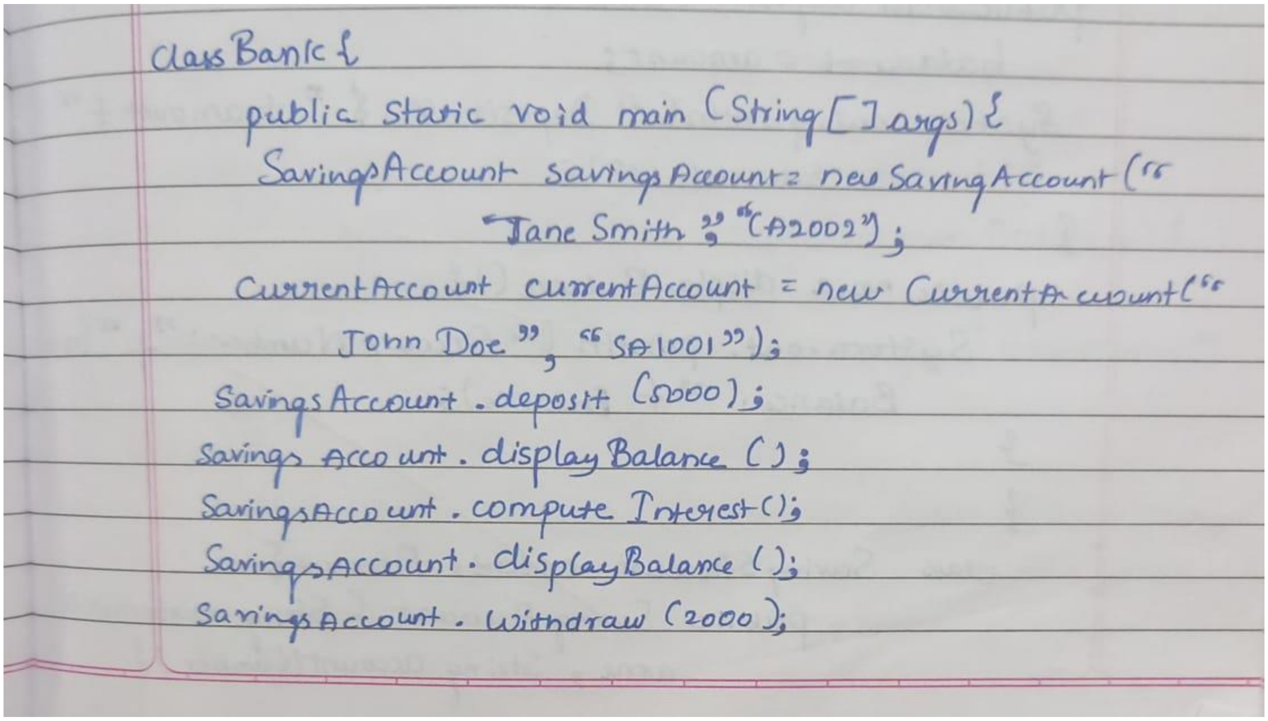

```
Class Bank {
    public static void main (String[] args) {
     SavingsAccount savingsAccount= new SavingAccount ("
              "Jane Smith", "CA2002");
   CurrentAccount currentAccount = new CurrentAccount("
          John Doe", "SA1001");
  SavingsAccount . deposit (5000);
 savings Account . displayBalance ();
 SavingsAccount . compute Interest();
 SavingsAccount . displayBalance ();
 SavingsAccount . withdraw (2000);
```

```
SavingsAccount. displayBalance();
currentAccount. deposit (8000);
currentAccount. displayBalance ();

currentAccount. withdraw(5000);
current account displayBalance ();
}
}
 Class Account {
   protected String CustomerName;
   protected String accountNumber;
    protect double balance;
    public Account ( String Customername, String accou
          t number ) {
  this. customerName = CustomerName;
  this. accountNumber = accountNumber;
   this. balance =0;
 }
 Public void deposit (double amount) {
     balance + = amount;
  System. out. println ("Deposit of $" + amount + "
              Successful");
 }
   public voia displayBalance (){
      System.out. println ("Account Number: " + "\n
           Balance: " + Balance);
    }
    }
     class SavingsAccount extends Account {
              public SavingAccount ( String customerN
                    ame, String accountNumber ){
```

```
Super ( customerName , accountNumber);
}
 public void computeInterest () {
     double interestRate = 0.05;
     double interest = balance * interestRate;
     balance += interest;
    System.out.println ( "Interest of $ " + interest + "
         computed and added to the balance ");
 }
 public void withdraw (double amount ) {
     if (balance >= amount) {
       balance -= amount;
    System.out.println ("Withdrawal of "+ amount +
    " Successful " );
 } else {
       system.out.println (" Insuffient funds for
withdrawal. ");
} } }
class currentAccount extends Account {
   private double minimumBalance = 1000;
Public CurrentAccount (String CustomerName, String
          accountNumber) {
Super (CustomerName, accountNumber);
}
public void withdraw (double amount ){
   If (balance - amount >= minimum Balance) {
       balance -= amount;
System.out.println (" withdrawal of " + amount "
                   Successful. ");
}else{
  System.out.println (" Insufficient funds. service charge
    applied. ");
imposeServiceCharge ();
 } }
```

O/P

```
Deposit of $ 5000.0 Successful
Account Number: CA 2002
Balance :5000.0
Interest of $ 250.0 computed & added to the
balance
Account Number : CA2002
Balance :5250.0
withdrawal of 2000 Successful
Account Number : CA 2002
Balance : 3250
Deposit of $ 8000. 0 Successful
Account Number : SA 1001
Balance : 8000
withdrawal of 5000.0 successful
Account Number : SA1001
Balance : 3000.0
```

→ Program continued

```
private void imposeServiceCharge() {
    double service Charge = 20;
balance -= ServiceCharge;
SOP (" Service Charge Of $" + Service charge + " impos
        ");
}
```

22/1/24

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) {

BANK

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) {

BANK

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
```