

9/1/24 Lab-5

- Q) Develop a Java program to create a bank that maintains two kinds of accounts.

```
import java.util.*;
class account {
    String customer_name;
    int acc_no;
    String type;
    double balance;
    account(String name, int acc, String type,
             double balance) {
        name = customer_name;
        acc_no = acc;
        this.type = type;
        this.balance = balance;
    }
    void deposit(double amount) {
        balance += amount;
    }
    void withdraw(double amount) {
        if ((balance - amount) > 0) {
            balance -= amount;
        }
        else {
            System.out.println("Insufficient balance");
        }
    }
    void display() {
        System.out.println("Name: " + customer_name +
            "acc. no" + "acc_no" + "Type: " + type +
            "Balance" + balance);
    }
}
```

```
class sav_acc extends account {  
    private static double rate = 2;  
    sav_acc (String name, int acc_no,  
            double balance) {  
        Super (name, acc_no, "Savings", balance);  
    }  
    void interest () {  
        balance += balance * (rate) / 100;  
        System.out.println ("balance" + balance);  
    }  
}
```

```
class cur_acc extends account {  
    cur_acc (String name, int acc_no,  
            double balance) {  
        Super (name, acc_no, "current", balance);  
    }  
    private double minbal = 250;  
    private double service charge = 100;  
    void checkmin () {  
        if (balance < minbal) {  
            System.out.println ("Insufficient  
            balance, service charges are added");  
            balance -= service charge;  
            System.out.println ("Balance is: " + balance);  
        }  
    }  
}
```

```
public class bank {  
    public static void main(String args[]) {  
        Scanner s = new Scanner(System.in);  
        System.out.println("Enter name");  
        String name = s.next();  
        System.out.println("Enter type");  
        String type = s.next();  
        System.out.println("Enter acc. no");  
        int acc_no = s.nextInt();  
        System.out.println("Enter initial balance");  
        double balance = s.nextDouble();  
        int ch;  
        double amt1, amt2;  
        account acc = new account(name, acc_no,  
            type, balance);  
        sav_acc sa = new sav_acc(name, acc_no,  
            balance);  
        cur_acc ca = new cur_acc(name, acc_no,  
            balance);  
        while (true) {  
            if (acc.type.equals("savings")) {  
                System.out.println("Menu: 1. Deposit 2.  
                Withdraw 3. Interest 4. Display");  
                System.out.println("Enter the choice");  
                ch = s.nextInt();  
                switch (ch) {  
                    {
```



```
case 1: System.out.println("Enter amt:");  
amt1 = s.nextInt();
```

```
sa.deposit(amt1);
```

```
break;
```

```
case 2: System.out.println("Enter amt:");
```

```
amt2 = s.nextInt();
```

```
sa.withdraw(amt2);
```

```
break;
```

```
case 3: sa.interest();
```

```
break;
```

```
case 4: sa.display();
```

```
break;
```

```
default: System.out.println("Invalid input");  
break;
```

```
}
```

```
else {
```

```
System.out.println("Menu 1. Deposit 2.  
Withdraw. 3. Display");
```

```
System.out.println("Enter choice");
```

```
ch = s.nextInt();
```

```
switch(ch)
```

```
{
```

```
case 1: System.out.println("Enter amt");
```

```
amt1 = s.nextInt();
```

```
sa.deposit(amt1);
```

```
break;
```

```
case 2: System.out.println("Enter amt");
```

```
amt2 = s.nextInt();
```

```
sa.withdraw(amt2);
```

```
break;
```

```
case 3: ca.display();
```

```
break;
```

```
case 4;
```

```
default: System.out.println("Invalid input");
```

```
System.exit(0);
```

```
{
```

```
}
```

```
}
```

```
}
```

```
}
```

O/p:

Enter name: Shreya

enter type: current

enter acc. number: 8288

Enter initial balance: 2000

Menu 1. deposit 2. withdraw 3. display

Enter choice: 1

Enter amount: 4000

Enter choice: 3

Name: shreya

type: current

acc. number: 8288

balance: 6000