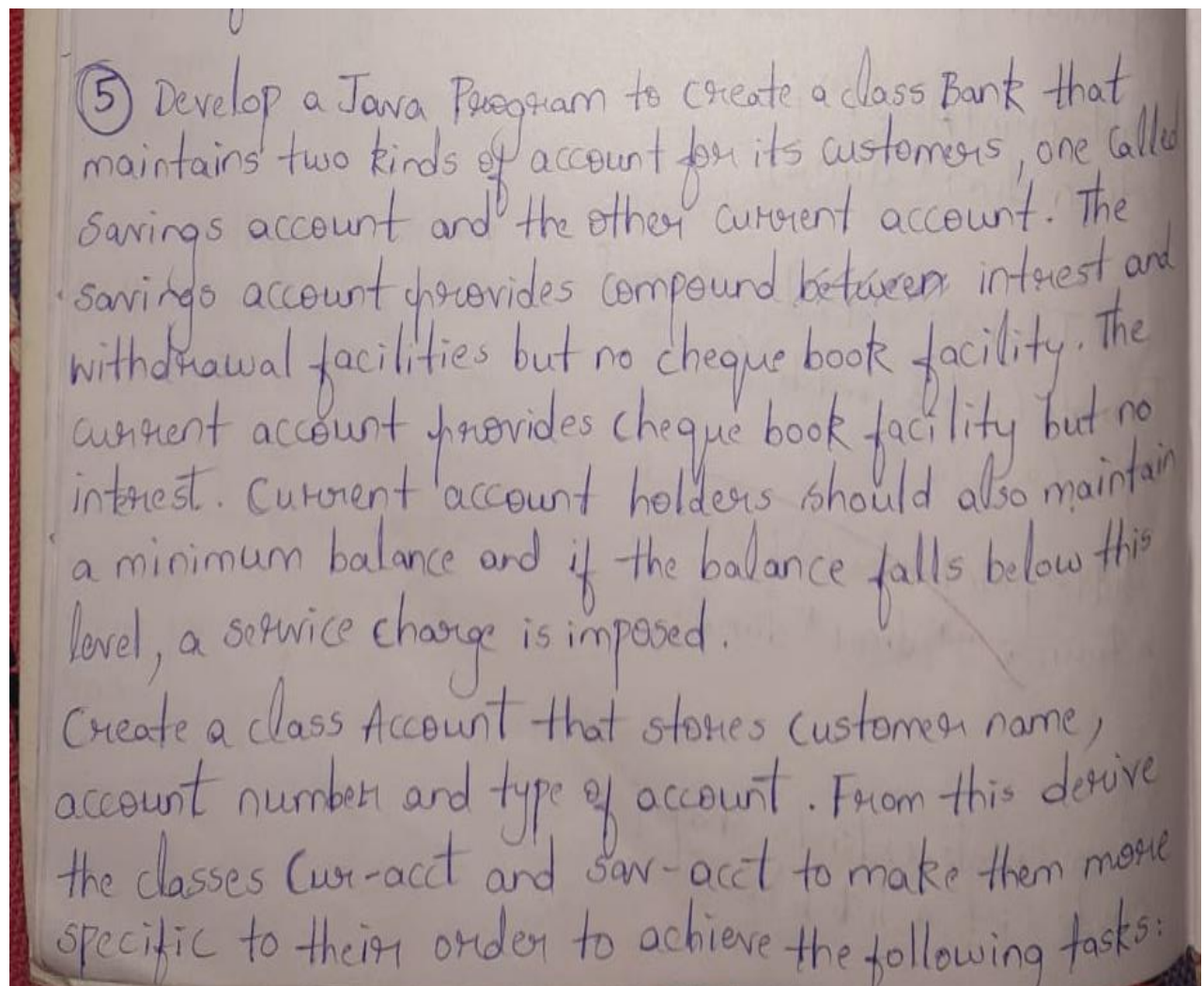


Program 5

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.



- Accept deposit from customer and update the balance
 - Display the balance
 - Compute and deposit interest
 - Permit withdrawal and update the balance
- Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;

abstract class Account {
    String customerName, accountNumber;
    double balance;

    Account(String customerName, String accountNumber,
            double initialBalance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }

    abstract void deposit(double amount);
    abstract void displayBalance();
    abstract void withdraw(double amount);
}

class SavAcct extends Account {
    double interestRate;

    SavAcct(String customerName, String accountNumber,
            double initialBalance) {
        super(customerName, accountNumber, initialBalance);
        this.interestRate = interestRate;
    }

    void deposit(double amount) {
        balance += amount;
    }

    void displayBalance() {
        System.out.println("Savings Balance: " + balance);
    }
}
```



```

void withdraw (double amount) {
    if (amount <= balance) balance -= amount;
}
void computeAndDepositInterest () {
    balance += balance * interestRate / 100;
}

```

```

class CurAct extends Account {
    static final double MIN_BALANCE = 1000, SERVICE_CHARGE = 50;
    CurAct (String customerName, String accountNumber,
            double initialBalance) {
        super (customerName, accountNumber, initialBalance);
    }
    void deposit (double amount) {
        balance += amount;
    }
    void displayBalance () {
        System.out.println ("Current Balance: " + balance);
    }
    void withdraw (double amount) {
        if (amount <= balance) {
            balance -= amount;
            if (balance < MIN_BALANCE) balance -= SERVICE_CHARGE;
        }
    }
}

```

```

class Bank {
    public static void main (String[] args) {
        Scanner scanner = new Scanner (System.in);
        System.out.println ("Enter account type (savings/
                                current):");

        String type = scanner.nextLine();
        String name = scanner.next();
        System.out.println ("Enter account name:");
        String name = scanner.nextLine();
        System.out.println ("Enter account Number:");
        String number = scanner.nextLine();
    }
}

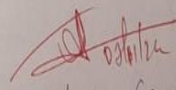
```

```

Account account;
if (type.equals ("savings")) {
    System.out.println ("Initial balance and interest rate:");
    account = new SavAct (name, number, scanner.nextDouble(),
                          scanner.nextDouble());
}
else {
    System.out.println ("Initial balance:");
    account = new CurAct (name, number, scanner.nextDouble());
}

while (true) {
    System.out.println ("1. Deposit 2. Display Balance
                        3. Withdraw 4. Interest 5. Exit");
    int choice = scanner.nextInt();
    switch (choice) {
        case 1: account.deposit (scanner.nextDouble());
                break;
        case 2: account.displayBalance();
                break;
        case 3: account.withdraw (scanner.nextDouble());
                break;
        case 4: if (account instanceof SavAct) {
                    ((SavAct) account).computeAndDepositInterest();
                }
                break;
        case 5:
                return;
    }
}

```

o/p: 
Enter account type (savings/current):
savings
Enter account name:
john
Enter account Number:
1235DFG

```
import java.util.Scanner;
```

```
abstract class Account {
```

```
    String customerName, accountNumber;
```

```
    double balance;
```

```
    Account(String customerName, String accountNumber, double initialBalance) {
```

```
        this.customerName = customerName;
```

```
        this.accountNumber = accountNumber;
```

```
        this.balance = initialBalance;
```

```
    }
```

```
    abstract void deposit(double amount);
```

```
    abstract void displayBalance();
```

```
    abstract void withdraw(double amount);
```

```
}
```

```
class SavAcct extends Account {
```

```
    double interestRate;
```

```
    SavAcct(String customerName, String accountNumber, double initialBalance, double  
interestRate) {
```

```
        super(customerName, accountNumber, initialBalance);
```

```
        this.interestRate = interestRate;
```

```
    }
```

```
    void deposit(double amount) {
```

```
        balance += amount;
```

```
    }
```

```
    void displayBalance() {
```

```

        System.out.println("Savings Balance: " + balance);
    }

    void withdraw(double amount) {
        if (amount <= balance) balance -= amount;
    }

    void computeAndDepositInterest() {
        balance += balance * interestRate / 100;
    }
}

class CurAcct extends Account {
    static final double MIN_BALANCE = 1000, SERVICE_CHARGE = 50;

    CurAcct(String customerName, String accountNumber, double initialBalance) {
        super(customerName, accountNumber, initialBalance);
    }

    void deposit(double amount) {
        balance += amount;
    }

    void displayBalance() {
        System.out.println("Current Balance: " + balance);
    }

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;

```

```

        if (balance < MIN_BALANCE) balance -= SERVICE_CHARGE;
    }
}
}

```

```

class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter account type (savings/current): ");
        String type = scanner.nextLine();

        System.out.println("Enter customer name: ");
        String name = scanner.nextLine();

        System.out.println("Enter account number: ");
        String number = scanner.nextLine();

        Account account;
        if (type.equals("savings")) {
            System.out.println("Initial balance and interest rate: ");
            account = new SavAcct(name, number, scanner.nextDouble(), scanner.nextDouble());
        } else {
            System.out.println("Initial balance: ");
            account = new CurAcct(name, number, scanner.nextDouble());
        }

        while (true) {
            System.out.println("\n1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit");
            int choice = scanner.nextInt();
            switch (choice) {

```

```

        case 1: account.deposit(scanner.nextDouble());
        break;

        case 2: account.displayBalance();
        break;

        case 3: account.withdraw(scanner.nextDouble());
        break;

        case 4: if (account instanceof SavAcct) ((SavAcct)
account).computeAndDepositInterest();
        break;

        case 5:
        return;
    }
}
}
}
}

```

```
D:\24BMSCE>javac Bank.java
```

```
D:\24BMSCE>java Bank
```

```
Enter account type (savings/current):
```

```
savings
```

```
Enter customer name:
```

```
anu rai
```

```
Enter account number:
```

```
123786645087301
```

```
Initial balance and interest rate:
```

```
5000
```

```
50
```

```
1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
```

```
1
```

```
200
```

```
1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
```

```
2
```

```
Savings Balance: 5200.0
```

```
1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
```

```
3
```

```
100
```

```
1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
```

```
4
```

```
1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
```

```
2
```

```
Savings Balance: 7650.0
```

```
1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
```

```
5
```