

09/01/24

## Lab program<sup>5</sup>

Date \_\_\_\_\_  
Page \_\_\_\_\_

Q) 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-Acc and Sav-Acc 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 ~~de~~ 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.io.BufferedReader;

Import java.io.IOException;

Import java.io.InputStreamReader;

class Account {

String customerName;

long accountNumber;

String accountType;

double balance;

public Account(String customerName,  
long accountNumber, String  
accountType){

this.customerName = customerName;

this.accountNumber = accountNumber;

this.accountType = accountType;

this.balance = 0.0;

public void deposit(double amount){

balance += amount;

System.out.println("Deposit  
successful. Updated Balance \$"  
+ balance);

public void displayBalance(){

System.out.println("Current  
balance: \$" + balance);

public void withdraw ( double amount ) {

if ( amount <= balance ) {

balance = amount ;

System.out.println (" withdrawal  
successful. Updated balance : " +

\$ " + balance );

} else {

System.out.println (" insufficient  
funds for withdrawal. " );

}

double minBalance = 500;

public Current ( String customerName,  
long accountNumber ) {

super ( customerName,  
accountNumber, " Current " );

}

public void checkMinBalance () {

if ( balance < minBalance ) {

double penalty = 50.0 ;

balance -= penalty ;

System.out.println (" Minimum  
Balance not maintained. " );

Penalty imposed. Updated  
balance: \$ " + balance );

class SavAcc extends Account {  
 double interestRate = 0.05;

public SavAcc (String customerName,  
 long accountNumber) {

super(customerName, accountNumber,  
 "Savings");  
 }

public void computeAndDepositInterest  
 (int timePeriod) {

double interest = balance \*  
 interestRate \* timePeriod;

balance += interest;

System.out.println("Interest  
 computed and deposited. Updated  
 balance \$" + balance);

public void withdraw (double amount)

if (amount <= balance) {  
 balance -= amount;

System.out.println("Withdrawal  
 successful. Update balance \$" +  
 balance);

public class Bank {

public static void main (String []  
 args) throws IOException

BufferedReader reader = new

BufferedReader (new InputStreamReader(Reader[System.out]));

System.out.print("Enter S for saving  
and C for current account : ");

String accountType = reader.readLine();

if (accountType.equalsIgnoreCase("C"))

System.out.print("Enter your name : ");

long accNo = Long.parseLong(reader.readLine());

Current currentAccount = new

Current(name, accNo);

System.out.println("Do you want to  
deposit? Type 'yes': ");

String depositChoice = reader.readLine();

if (depositChoice.equalsIgnoreCase("yes")) {

System.out.println("Enter the amount  
to deposit.");

double depositAmount = Double.

parseDouble(reader.  
readLine());

currentAccount. deposit (depositAmount);  
}

System.out.println ("Do you want to  
withdraw? Type 'yes': ");

String withdrawChoice = reader.readLine();

if (withdrawChoice.equalsIgnoreCase  
("yes")) {

System.out.println ("Enter the amount  
to withdraw: ");

double withdrawAmount = Double.  
parseDouble (reader.readLine());

currentAccount.withdraw (withdraw  
Amount);

} else if  
(accountType.equalsIgnoreCase ("S")) {

System.out.println ("Enter your name");

String name = reader.readLine();

System.out.println ("Enter your account  
number");

long accNo = Long.parseLong (reader.  
readLine());

SavAcc savingsAccount = new SavAcc  
(name, accNo);

System.out.println("Do you want  
to deposit? Type 'yes':");

String depositChoice = reader.readLine();

if (depositChoice.equals("yes")) {

System.out.println("Enter the amount  
to deposit:");

double depositAmount = Double.parseDouble(  
Double(reader.readLine()));

SavingsAccount.deposit(deposit  
Amount);

System.out.println("Enter the time  
period (in years) the money is kept:");

int timePeriod = Integer.parseInt(  
reader.readLine());

SavingsAccount.computeAndDepositInterest  
(time period);

System.out.println("Do you want  
to withdraw? Type 'yes':");

String withdrawChoice = reader.readLine();

{  
if withdrawChoice.equalsIgnoreCase("yes") {

System.out.println("Enter the amount to withdraw: ");

double withdrawAmount = Double.parseDouble(reader.readLine());

savingsAccount.withdraw(withdrawAmount);

savingsAccount.displayBalance();

} else {

System.out.println("Invalid account type. Please enter 'c' for current or 's' for savings account.");

}

reader.close();

}

~~Output: + (Savings)~~

→ Enter s for saving and c for current account: s

Enter your name: Setu

Enter your account number: 123

Do you want to deposit? Type 'yes':  
Yes

Enter the amount to deposit: 600

Deposit successful. Updated balance: \$600.

Enter the time period (in years) the  
money is kept: 3  
Interest computed and deposited,  
Updated balance: \$ 690.0

Do you want to withdraw? Type 'yes': Yes

Enter the amount to withdraw: 600

Withdraw successful. Updated balance:

Current balance: \$ 90.0

→ (Current)

Enter S for saving and C for current  
account: C

Enter your name: Selvi

Enter your account number: 1234

Do you want to deposit? Type 'yes': Yes

Enter the amount to deposit: \$4000.0

Deposit successful. Updated balance: \$ 4000.0

Do you want to withdraw? Type 'yes':  
Yes

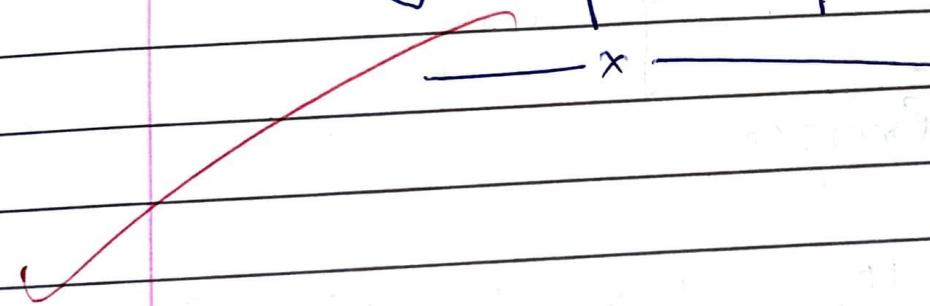
Enter the amount to withdraw: 3750

Withdrawal successful. Updated balance:  
\$250.0

Current balance: \$250.0

Minimum balance not maintained.

Penalty imposed. Updated balance: \$200.0



01/06/2024  
23