Experiment 9 – Thread Programming Simulate a bank account supporting opening/closing, withdrawals, and deposits of money. Treat them as concurrent transactions. A bank account can be accessed in multiple ways. Clients can make deposits and withdrawals. Create an account that can be accessed from multiple threads. It should be possible to close an account; operations against a closed account must fail.

import java.util.Scanner;

import java.util.NoSuchElementException;

class Account {

 float balance;

 Account(float amt){

    this.balance=amt;

 }

 public boolean iswithdrawable(float w)

 {

     if(this.balance>w)

        return true;

     else

        return false;

 }

 public float withdraws(float w)

 {

     if(iswithdrawable(w))

        this.balance=this.balance-w;

     else

         System.out.println("Insufficient Balance");

    return this.balance;

 }

 public float deposits(float d)

 {

     this.balance=this.balance+d;

     return this.balance;

 }

}

class Withdraw implements Runnable

{

    Account A;

    Withdraw(Account a)

    {

        this.A=a;

    }

    public void run()

    {

        try{

            synchronized(A){

            Scanner sc=new Scanner(System.in);

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

            float w=sc.nextFloat();

            w=this.A.withdraws(w);

            System.out.println("Your current balance is "+w);}

            }

        catch(NoSuchElementException e)

        {

            System.out.println(e);

        }

    }

}

class Deposit implements Runnable{

    Account A;

    Deposit(Account a)

    {

        this.A=a;

    }

    public void run()

    {

        try{

        synchronized(A)

        {

            Scanner sc=new Scanner(System.in);

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

            float d=sc.nextFloat();

            d=this.A.deposits(d);

            System.out.println("your balance is "+d);

        }}

        catch(NoSuchElementException e)

        {

            System.out.println(e);

        }

    }

}

public class exp9 {

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the amount to create an account");

        float m=sc.nextFloat();

        Account A=new Account(m);

        Thread t1=new Thread(new Withdraw(A));

        Thread t2=new Thread(new Deposit(A));

        t1.start();

        t2.start();

        //if you want to close an account then

        // t1.stop();

        // t2.stop();

    }

}

OUTPUT:-

