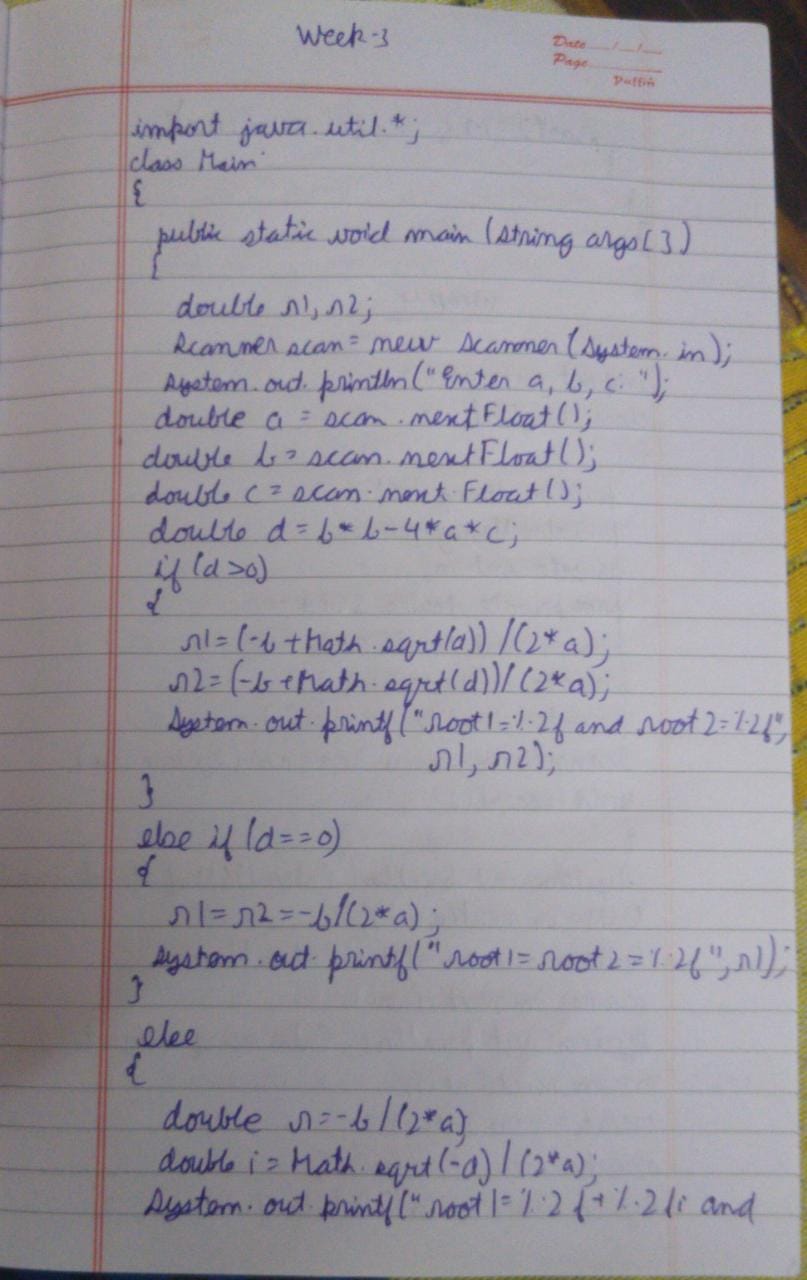
**LAB1**:

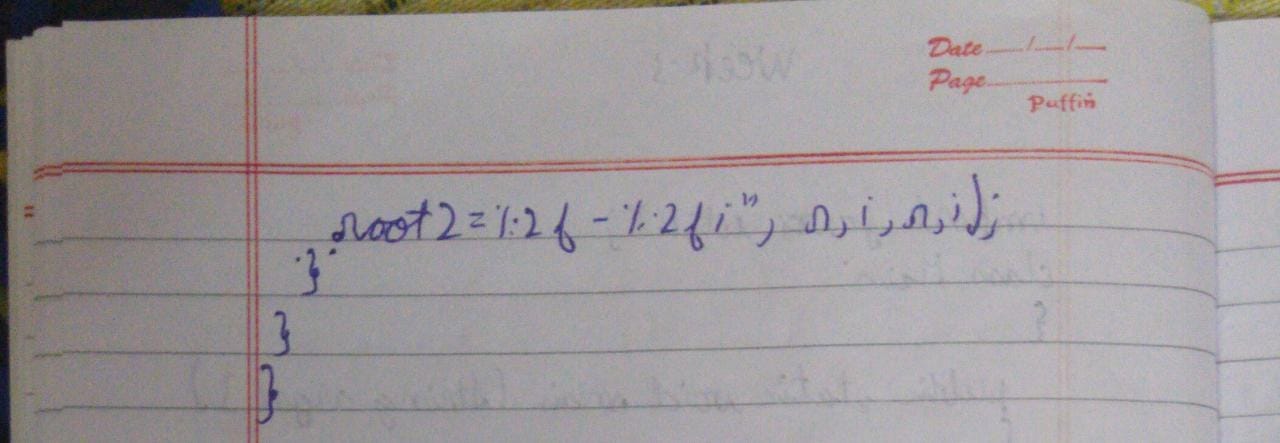
Develop a Java program that prints all real solutions to the quadratic equation ax2 +bx+c = 0.

Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a

message stating that there are no real solutions.

**OBSERVATION:**





**SOURCE CODE:**

import java.util.\*;

class Main

{

public static void main(String args[])

{

double r1,r2;

Scanner scan = new Scanner(System.in);

System.out.println("Enter the coefficients a, b, c: ");

double a = scan.nextFloat();

double b = scan.nextFloat();

double c = scan.nextFloat();

double d=b\*b-4\*a\*c;

if(d>0)

{

r1=(-b+Math.sqrt(d))/(2\*a);

r2=(-b-Math.sqrt(d))/(2\*a);

System.out.printf("root1=%.2f and root2=%.2f",r1,r2);

}

else if(d==0)

{

r1=r2=-b/(2\*a);

System.out.printf("root1=root2=%.2f",r1);

}

else

{

double r=-b/(2\*a);

double i=Math.sqrt(-d)/(2\*a);

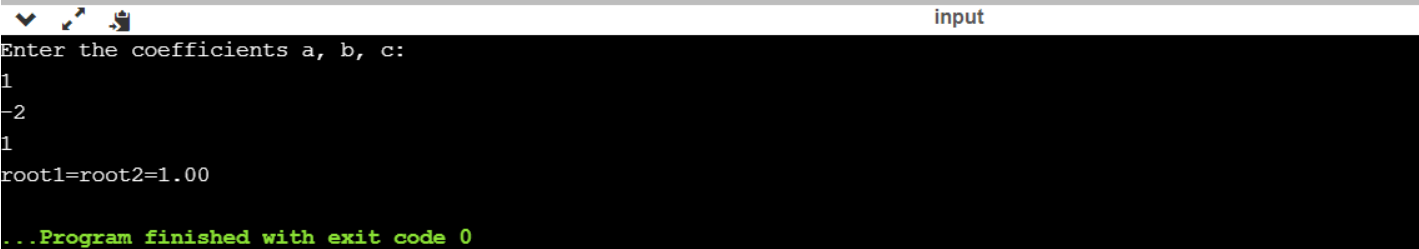
System.out.printf("root1=%.2f+%.2fi and root2=%.2f-%.2fi",r,i,r,i);

}

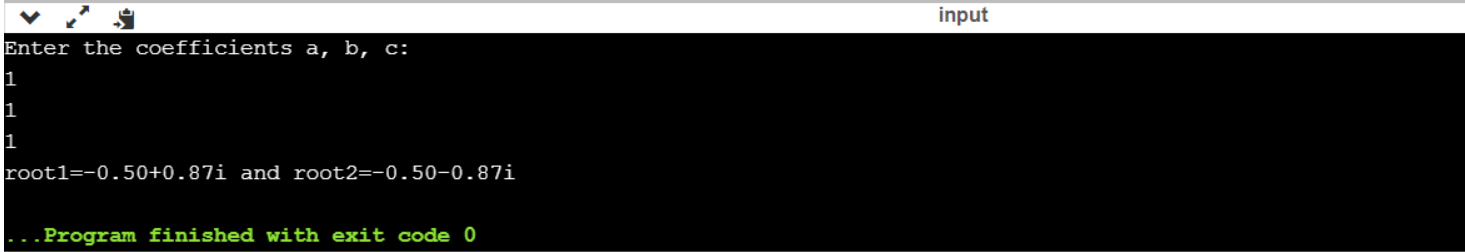
}

}

**OUTPUT:**

****

****

****

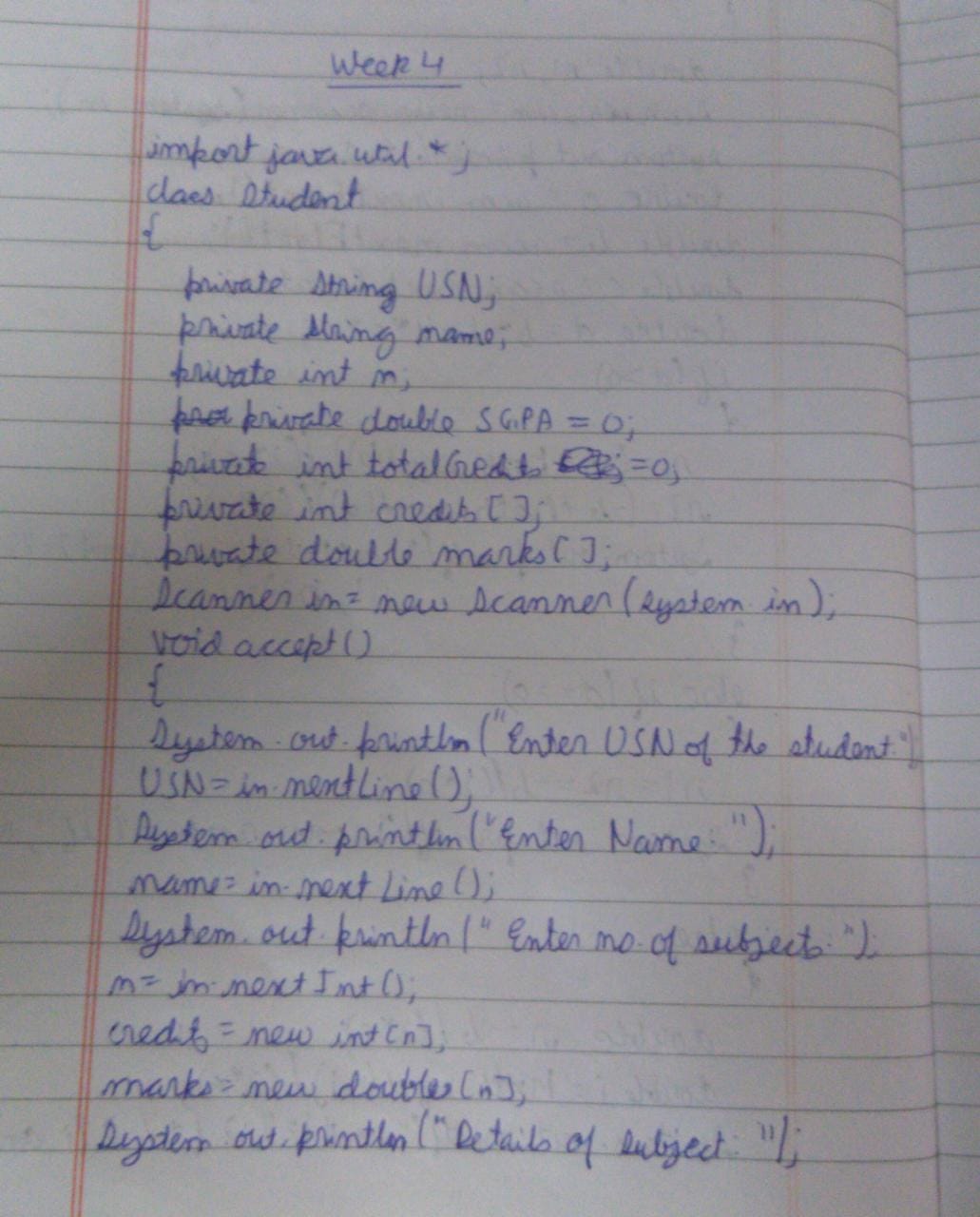
**LAB2:**

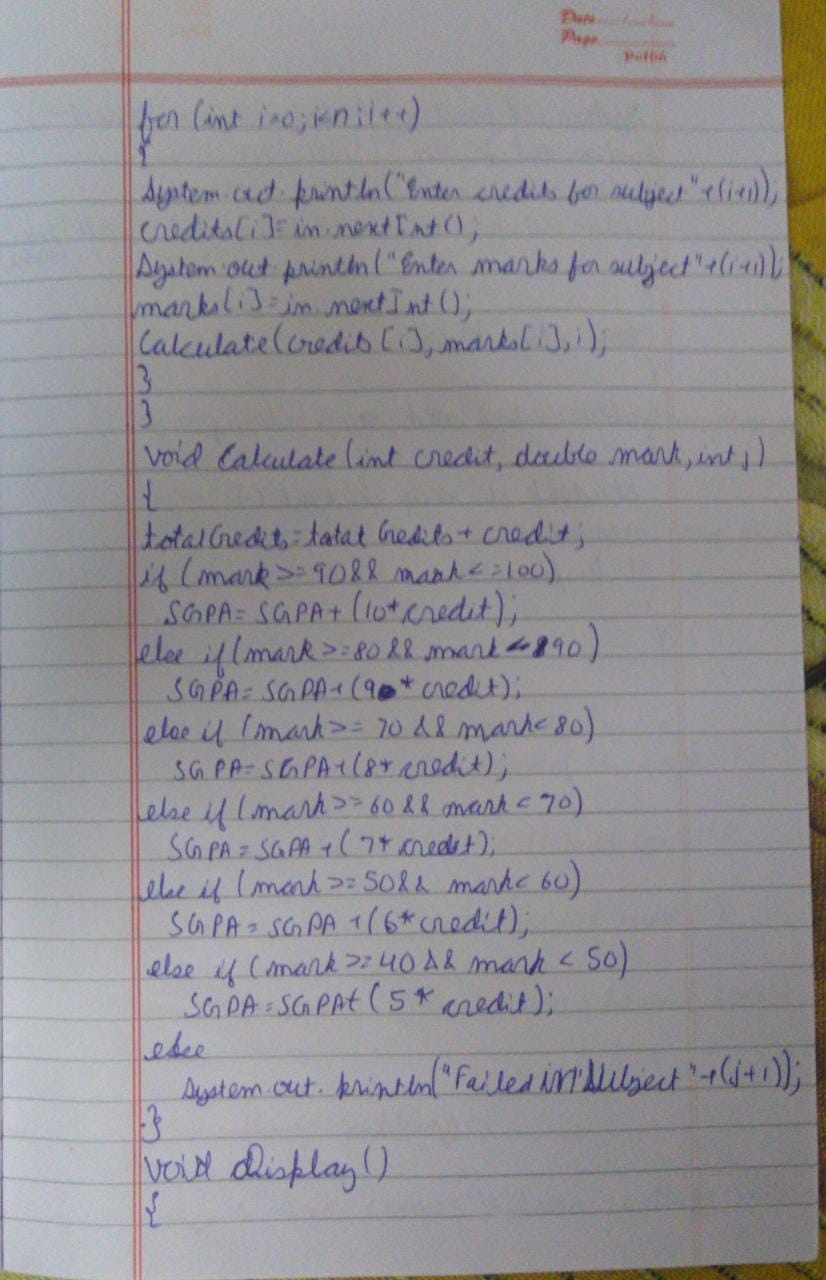
Develop a Java program to create a class Student with members usn, name, an array

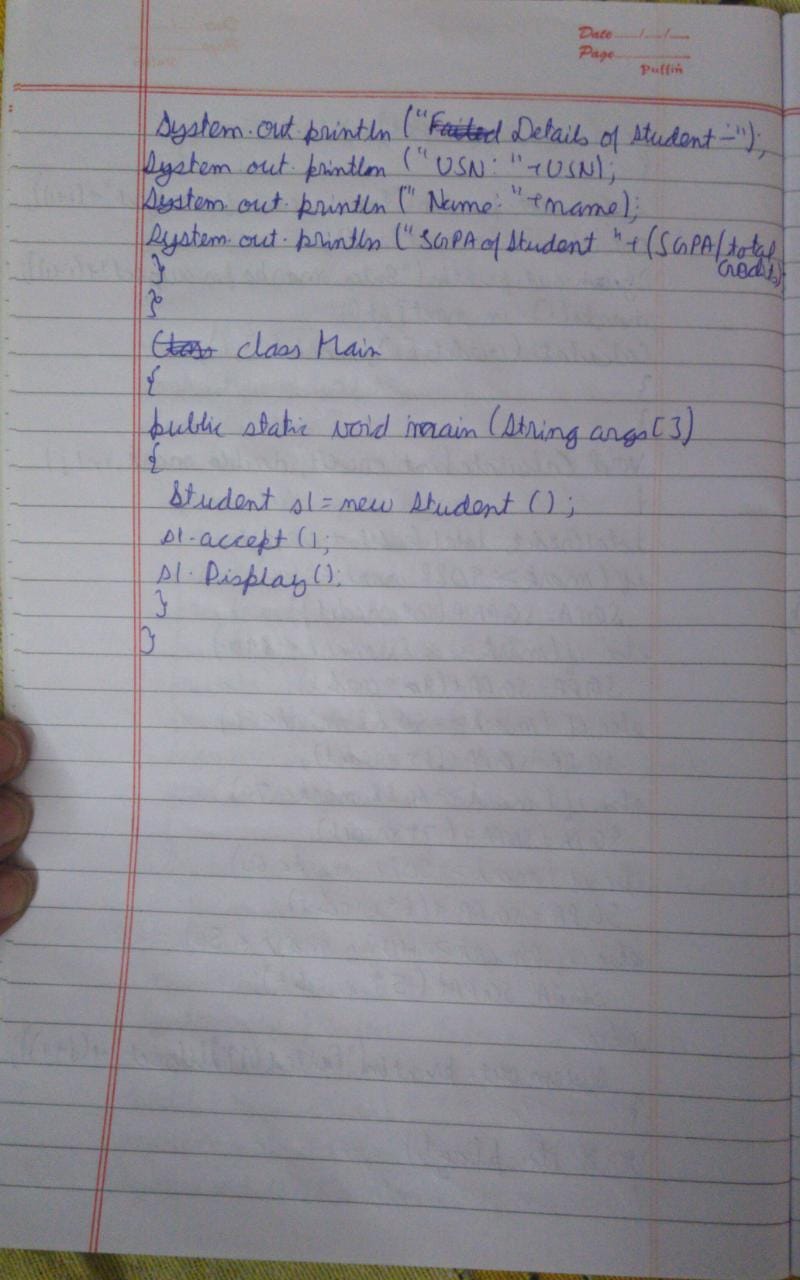
credits and an array marks. Include methods to accept and display details and a method to

calculate SGPA of a student.

**OBSERVATION:**

****

****

****

**SOURCE CODE:**

import java.util.\*;

class Student

{

private String USN;

private String name;

private int n;

private double SGPA = 0;

private int totalCredits = 0;

private int credits[];

private double marks[];

Scanner in = new Scanner(System.in);

void accept()

{

System.out.println("Enter USN of the student");

USN = in.nextLine();

System.out.println("Enter Name of the student");

name = in.nextLine();

System.out.println("Enter number of subjects");

n = in.nextInt();

credits = new int[n];

marks = new double[n];

System.out.println("Details of the subjects:");

for(int i=0;i<n;i++)

{

System.out.println("Enter credits for subject "+(i+1));

credits[i] = in.nextInt();

System.out.println("Enter marks for subject "+(i+1));

marks[i] = in.nextInt();

Calculate(credits[i],marks[i],i);

}

}

void Calculate(int credit,double mark,int j)

{

totalCredits = totalCredits + credit;

if(mark>=90&&mark<=100)

SGPA = SGPA + (10\*credit);

else if(mark>=80 && mark<90)

SGPA = SGPA + (9\*credit);

else if(mark>=70&&mark<80)

SGPA = SGPA + (8\*credit);

else if(mark>=60&&mark<70)

SGPA = SGPA + (7\*credit);

else if(mark>=50 && mark<60)

SGPA = SGPA + (6\*credit);

else if(mark>=40&&mark<50)

SGPA = SGPA + (5\*credit);

else

System.out.println("Failed in Subject "+(j+1));

}

void Display()

{

System.out.println("Details of the Student");

System.out.println("USN: "+USN);

System.out.println("Name :"+name);

System.out.println("SGPA of Student "+(SGPA/totalCredits));

}

}

class Main {

public static void main(String args[]) {

Student s1 = new Student();

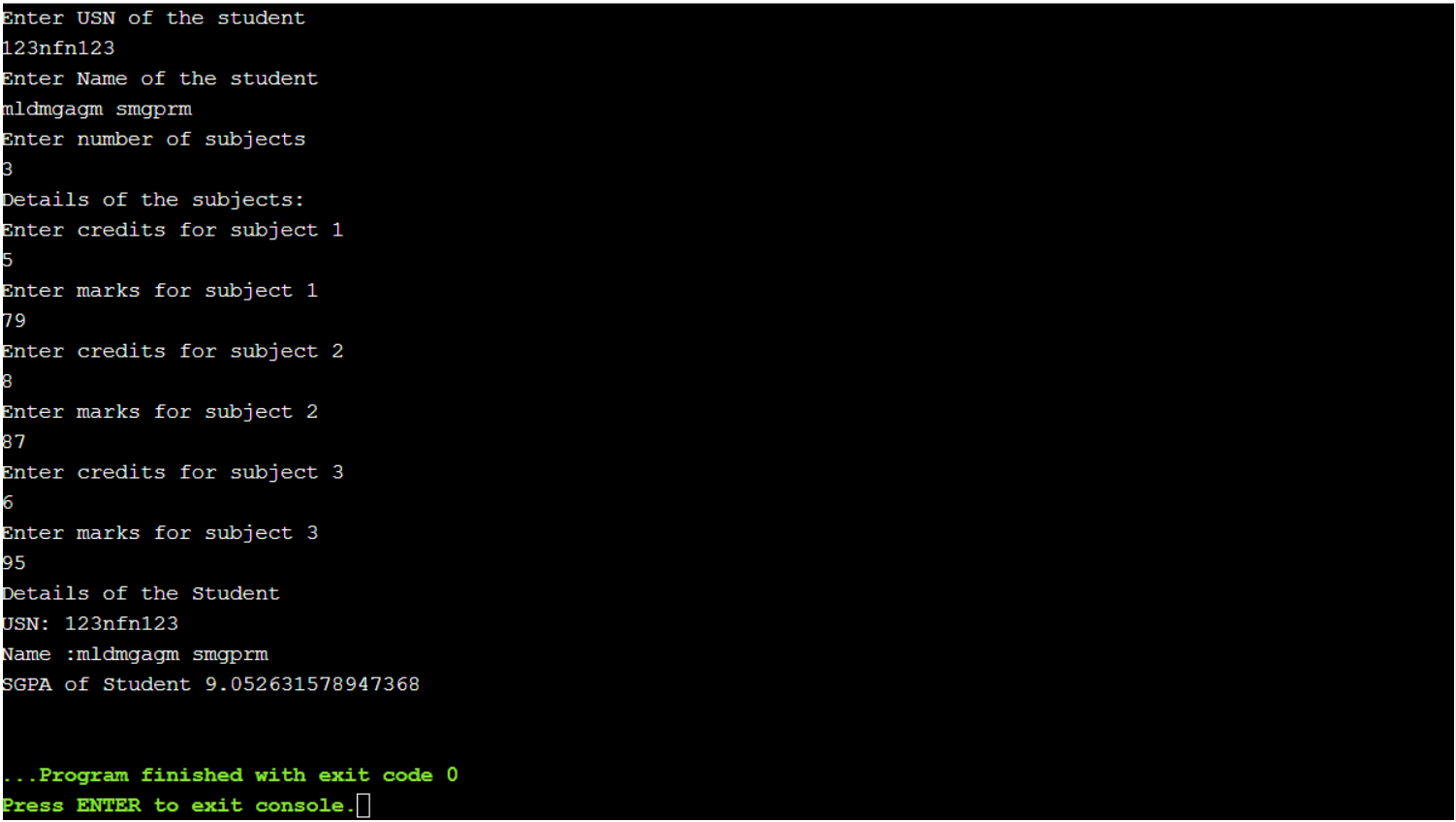
s1.accept();

s1.Display();

}

}

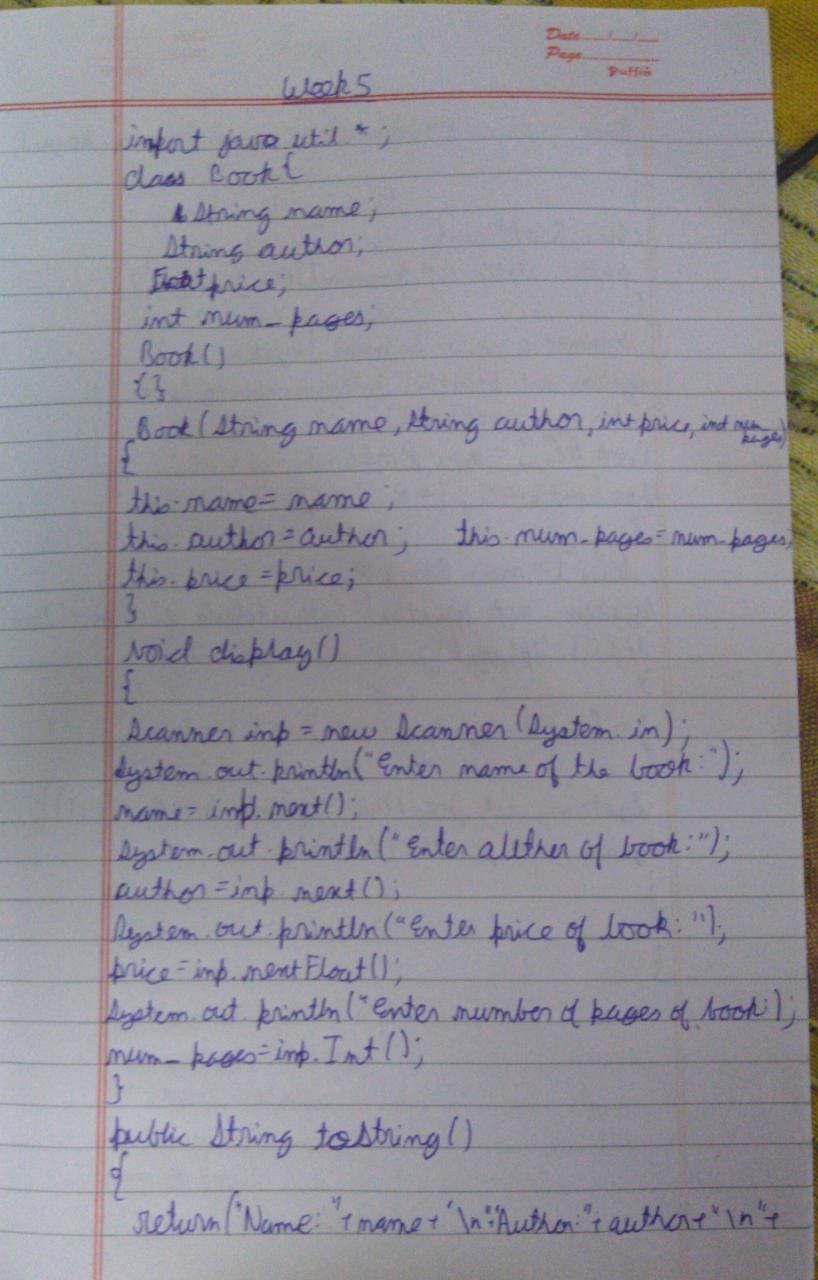
**OUTPUT:**

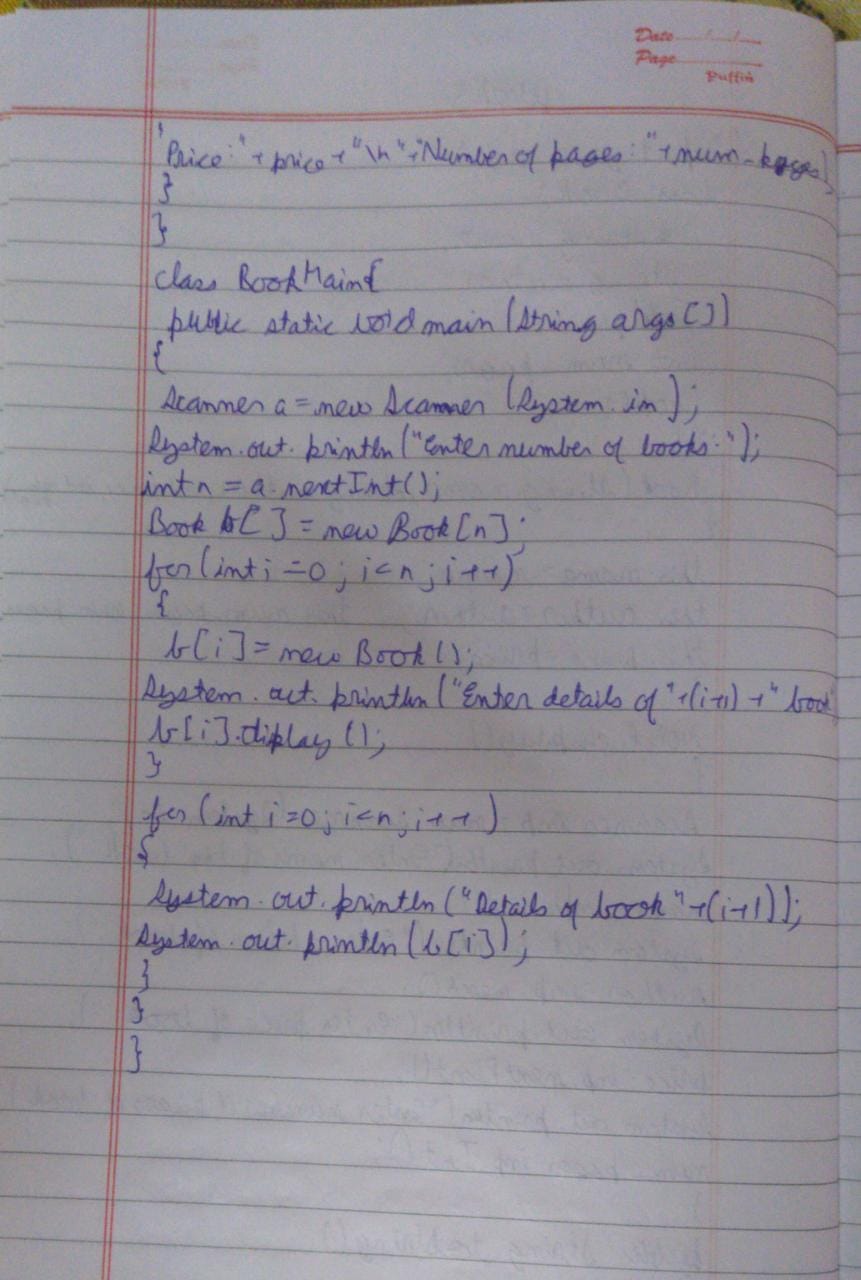
****

**LAB3:**

Create a class Book which contains four members: name, author, price, num\_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

**OBSERVATION:**

****

****

**SOURCE CODE:**

import java.util.\*;

class Book {

String name;

String author;

float price;

int num\_pages;

Book()

{}

Book(String name,String author,int price,int num\_pages)

{

this.name=name;

this.author=author;

this.price=price;

this.num\_pages=num\_pages;

}

void display()

{

Scanner inp=new Scanner(System.in);

System.out.println("Enter name of book:");

name=inp.next();

System.out.println("Enter author of book:");

author=inp.next();

System.out.println("Enter price of book:");

price=inp.nextFloat();

System.out.println("Enter number of pages of book:");

num\_pages=inp.nextInt();

}

public String toString()

{

return ("Name: "+name + "\n" + "Author: "+author + "\n" + "Price: "+price + "\n" +"Number of pages: "+num\_pages );

}

}

class BookMain {

public static void main(String args[])

{

Scanner a=new Scanner(System.in);

System.out.println("Enter the number of books:");

int n=a.nextInt();

Book b[]=new Book[n];

for(int i=0;i<n;i++)

{

b[i]=new Book();

System.out.println("Enter the details of "+(i+1)+" book");

b[i].display();

}

for(int i=0;i<n;i++)

{

System.out.println("Details of book "+(i+1));

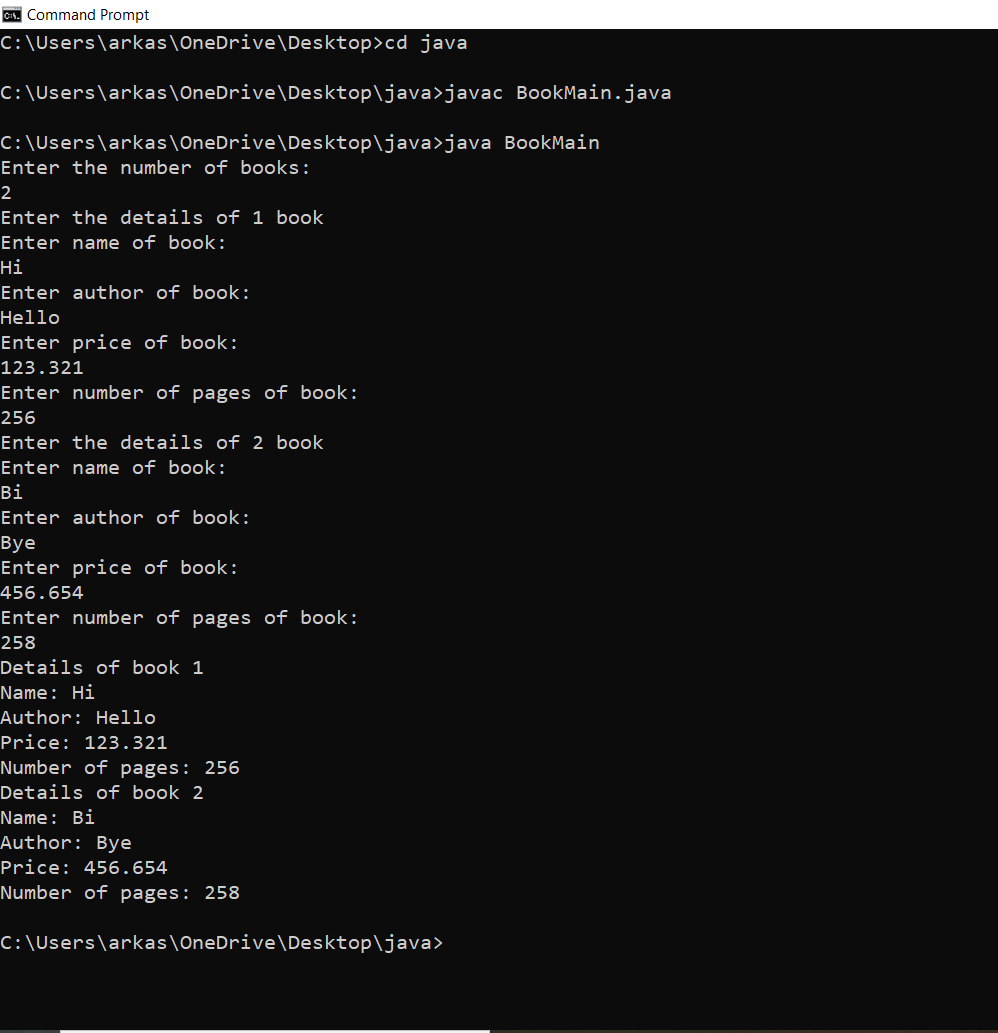
System.out.println(b[i]);

}

}

}

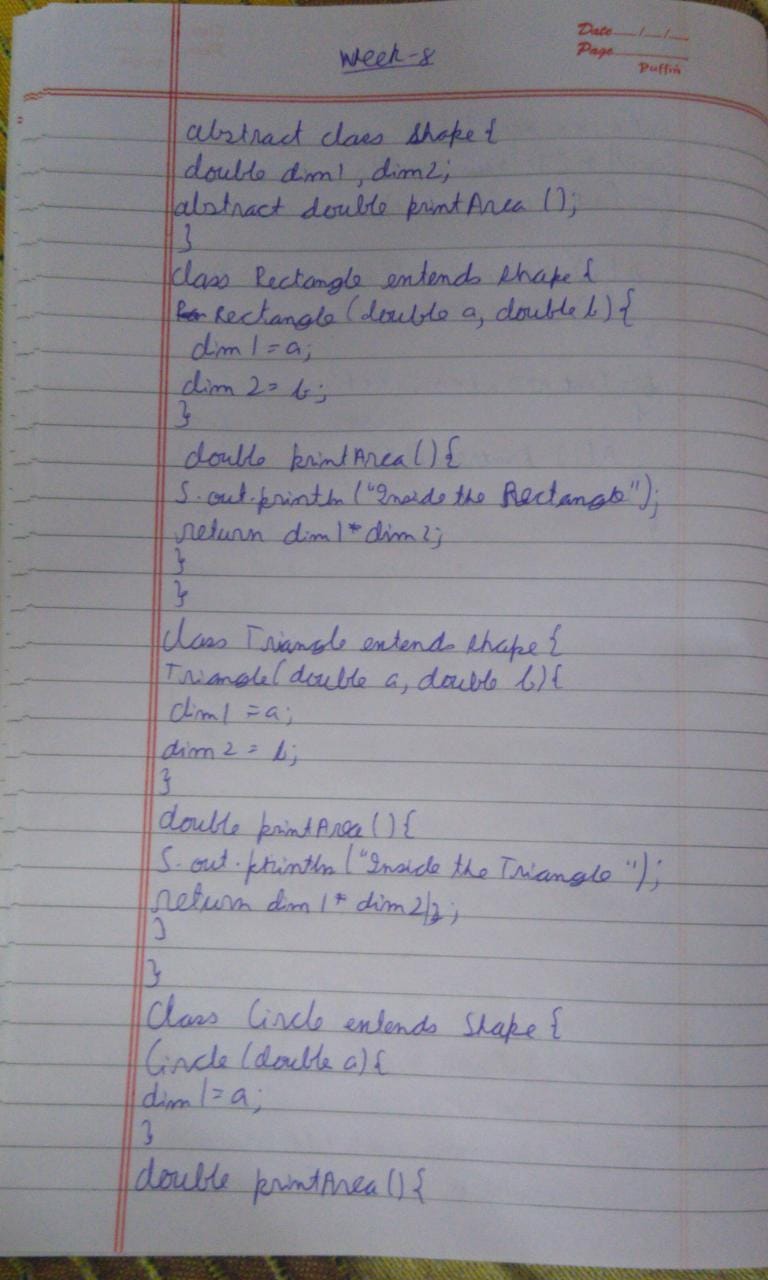
**OBSERVATION:**

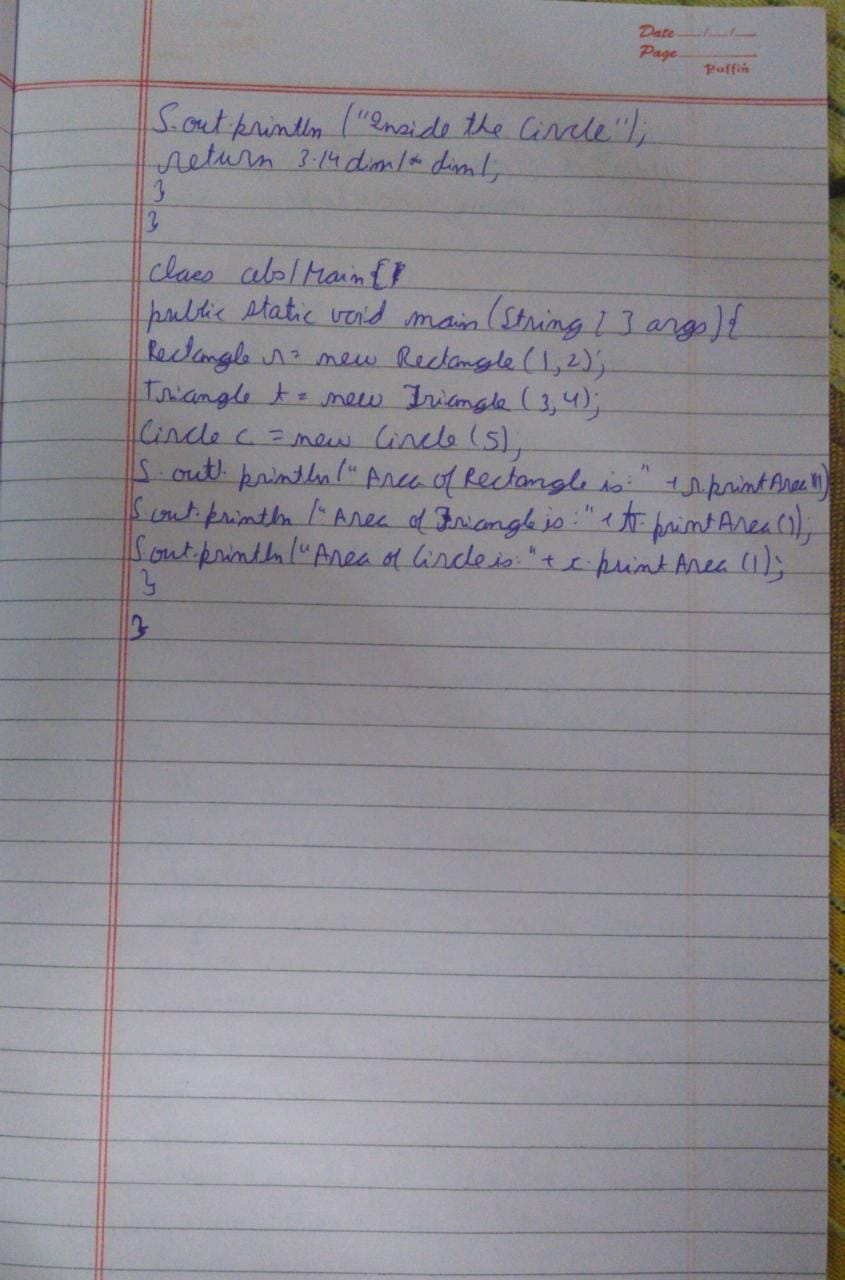
****

**LAB4:**

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

**OBSERVATION:**

****

****

**SOURCE CODE:**

import java.util.\*;

abstract class Shape{

double dim1,dim2;

abstract double printArea();

}

class Rectangle extends Shape{

Rectangle(double a, double b){

dim1 = a;

dim2 = b;

}

double printArea(){

System.out.println("Inside the Rectangle");

return dim1\*dim2;

}

}

class Triangle extends Shape{

Triangle(double a, double b){

dim1 = a;

dim2 = b;

}

double printArea(){

System.out.println("Inside the Triangle");

return dim1\*dim2/2;

}

}

class Circle extends Shape{

Circle(double a){

dim1 = a;

}

double printArea(){

System.out.println("Inside the Circle");

return 3.14\*dim1\*dim1;

}

}

class Abs{

public static void main(String args[]){

double rl,rb,th,tb,cr;

Scanner ss = new Scanner(System.in);

System.out.println("Enter length and breadth of rectangle: ");

rl = ss.nextDouble();

rb = ss.nextDouble();

System.out.println("Enter height and base length of triangle: ");

th = ss.nextDouble();

tb = ss.nextDouble();

System.out.println("Enter radius of circle: ");

cr = ss.nextDouble();

Rectangle r = new Rectangle(rl,rb);

Triangle t = new Triangle(th,tb);

Circle c = new Circle(cr);

System.out.println("Area of Rectangle is:" +r.printArea());

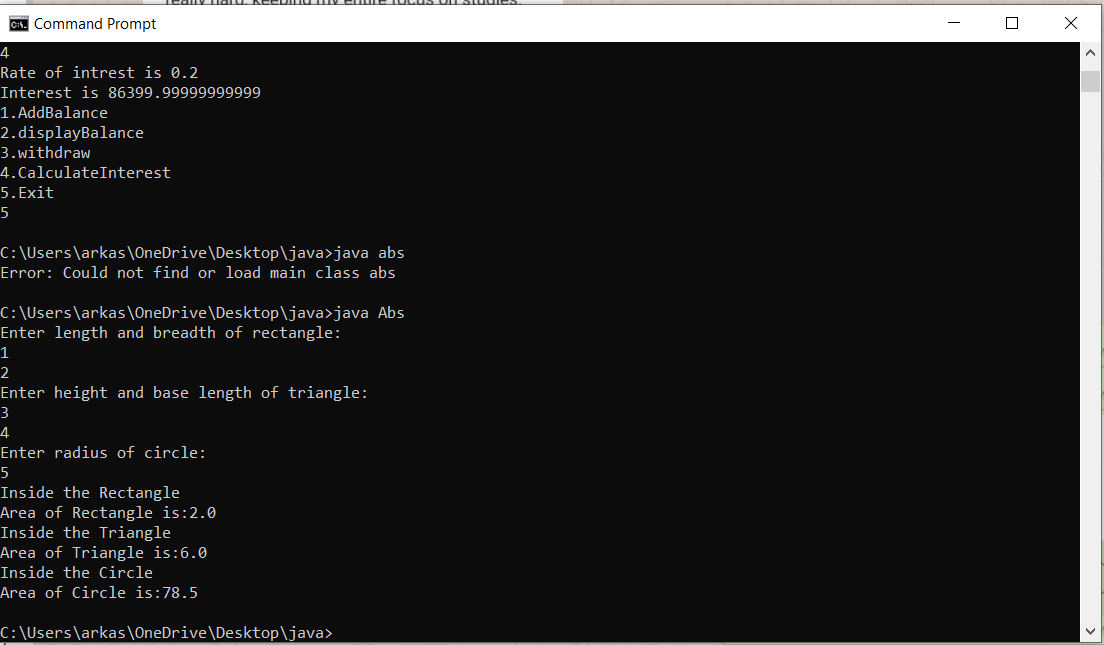
System.out.println("Area of Triangle is:" +t.printArea());

System.out.println("Area of Circle is:" +c.printArea());

}

}

**OUTPUT:**

****

**LAB5:**

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 Curr-acct and Sav-acct to make them more specific to

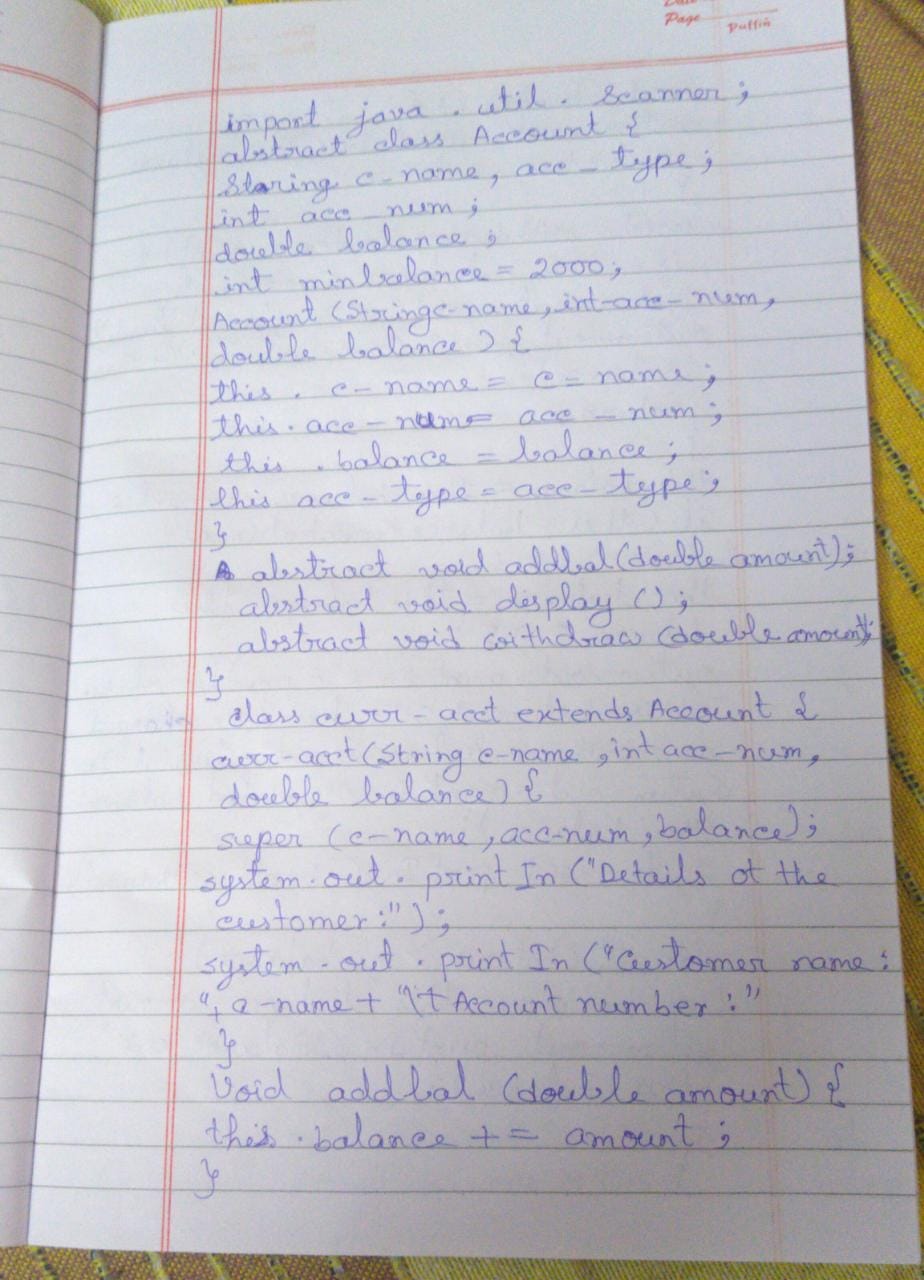
their requirements. Include the necessary methods in order to achieve the following tasks: •

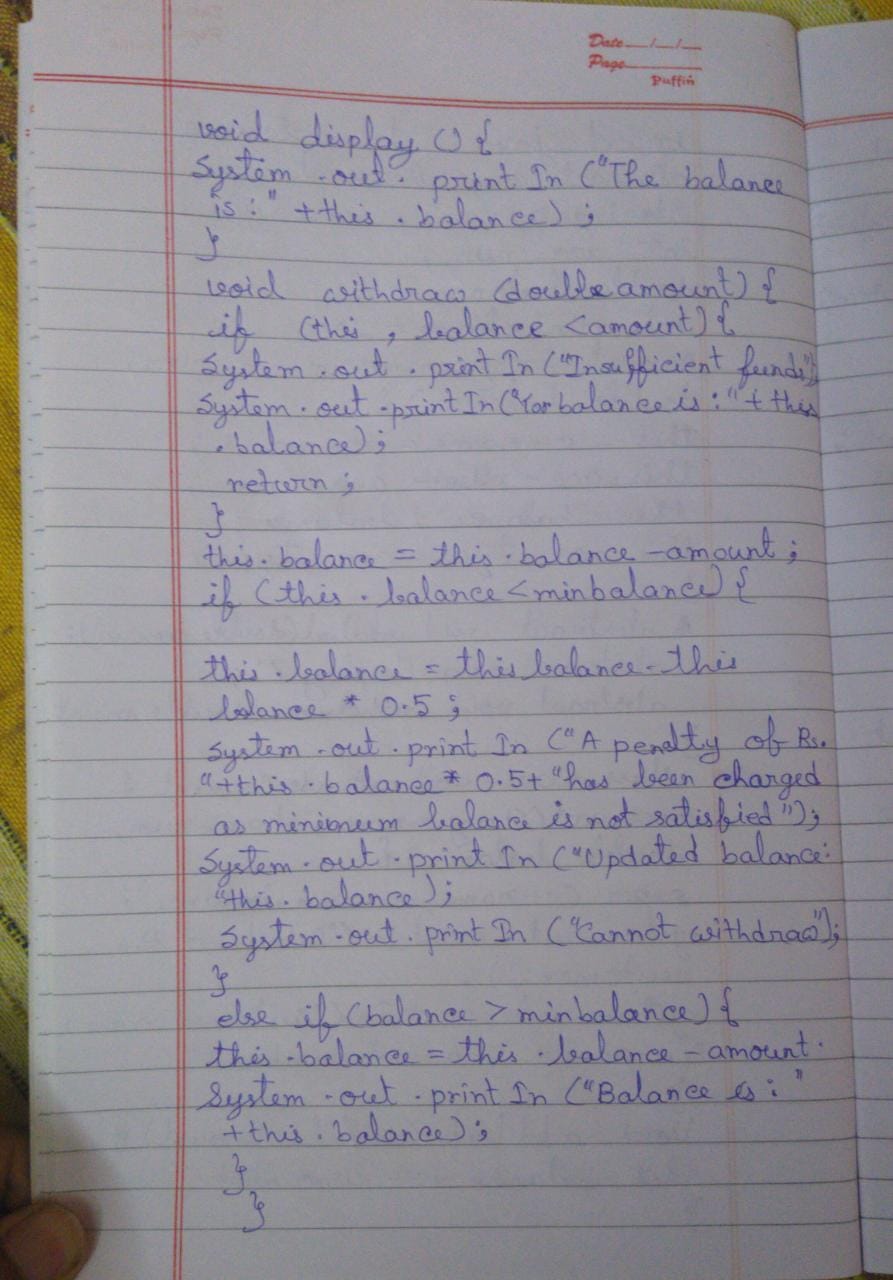
Accept deposit from customer and update the balance. • Display the balance. • Compute and

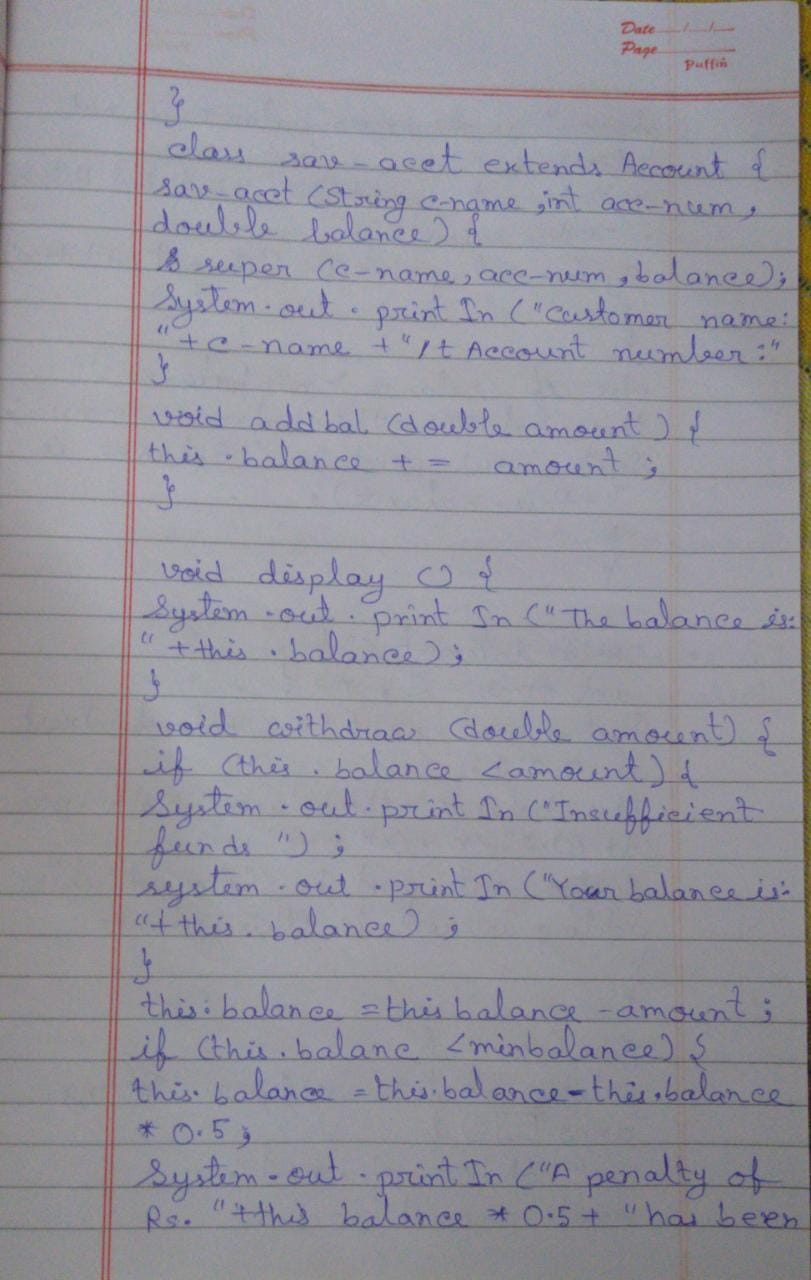
deposit interest • Permit withdrawal and update the balance • Check for the minimum balance,

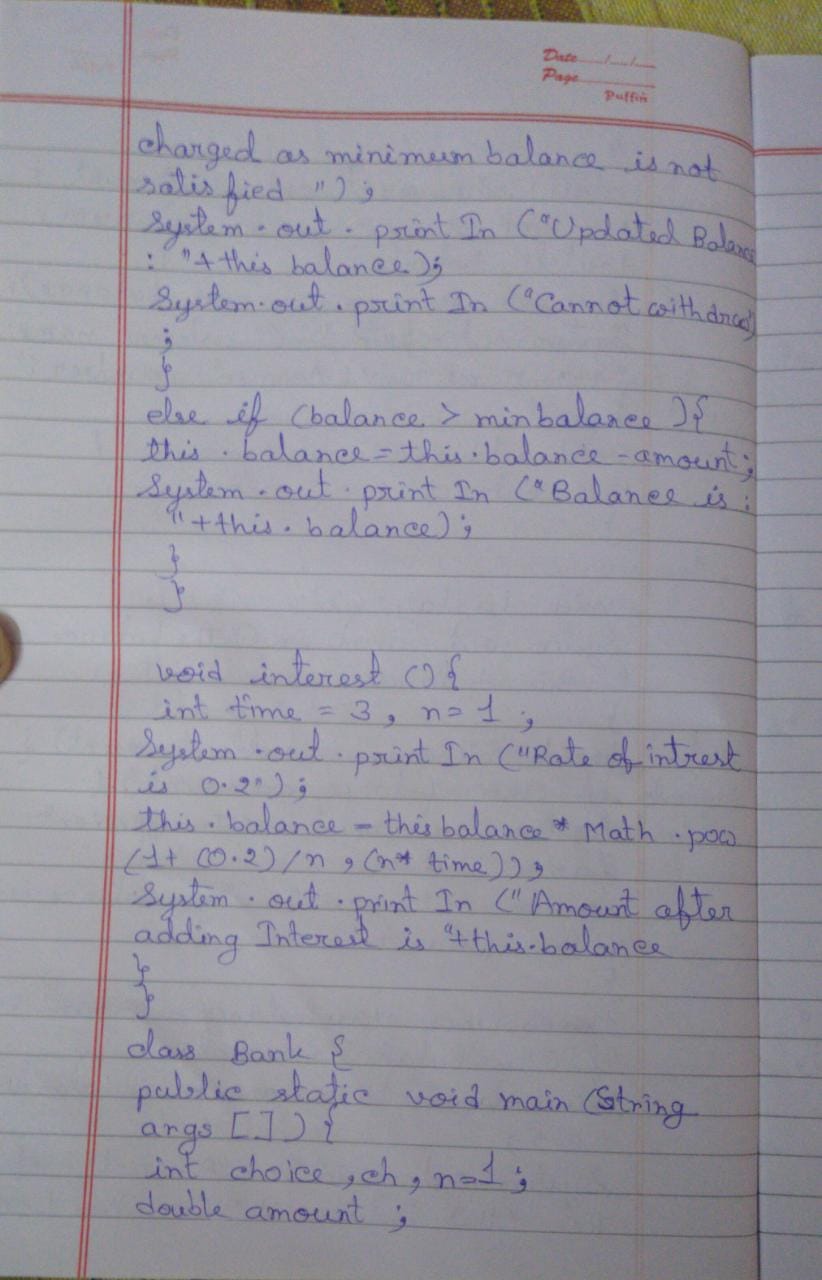
impose penalty if necessary and update the balance

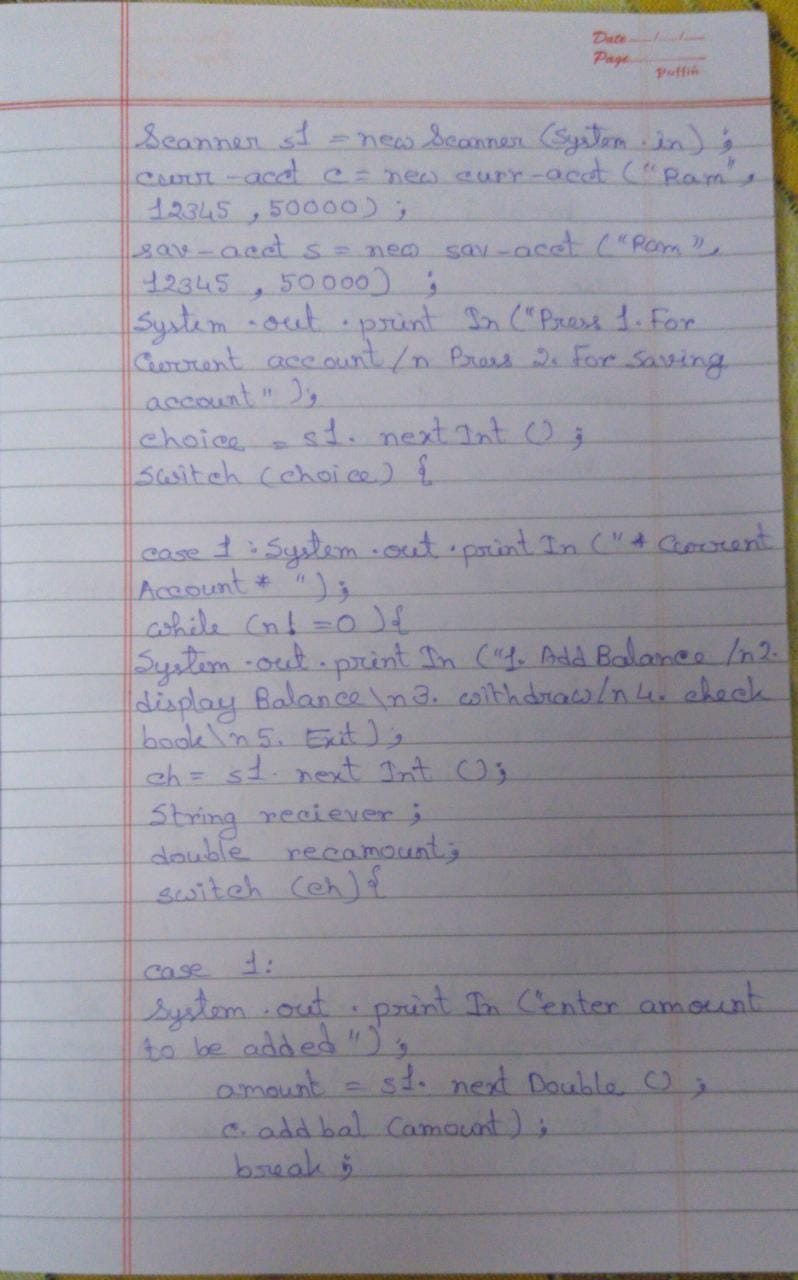
**OBSERVATION:**

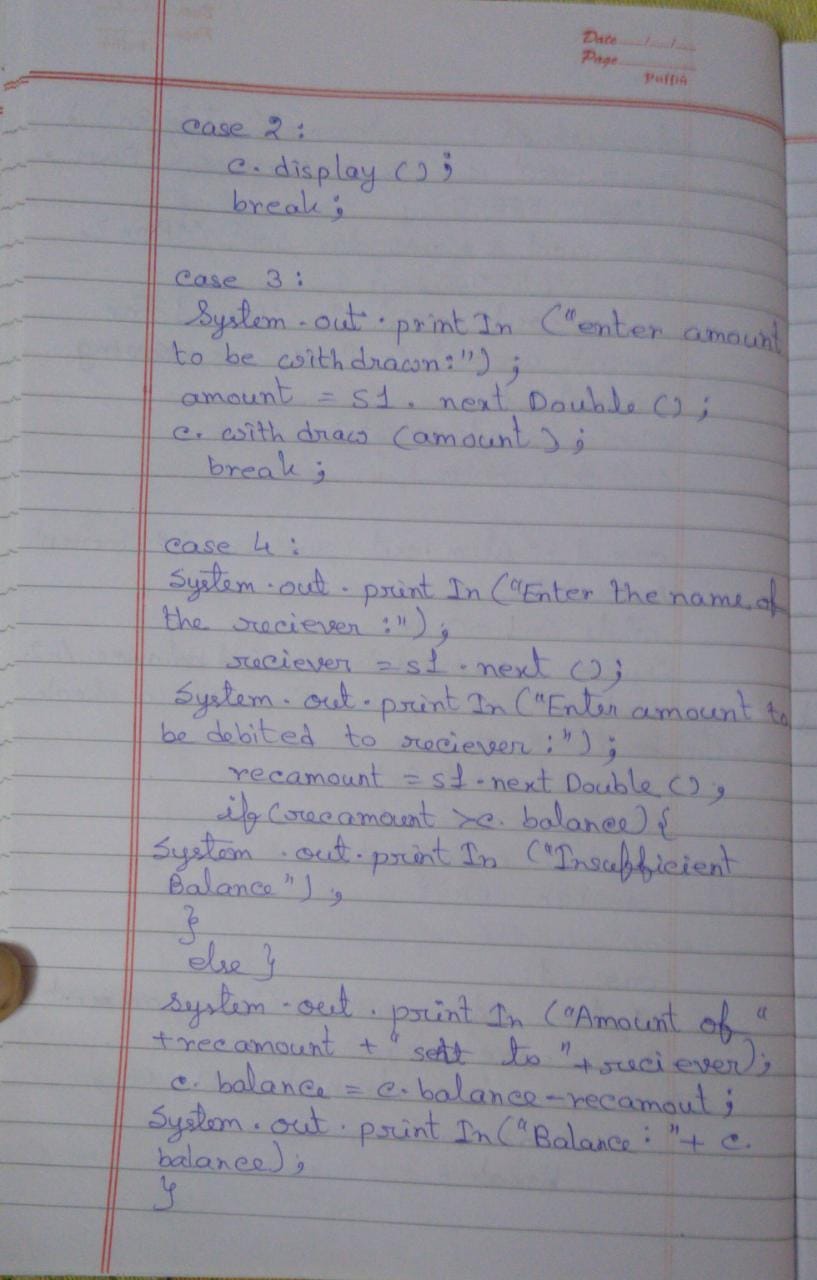
****

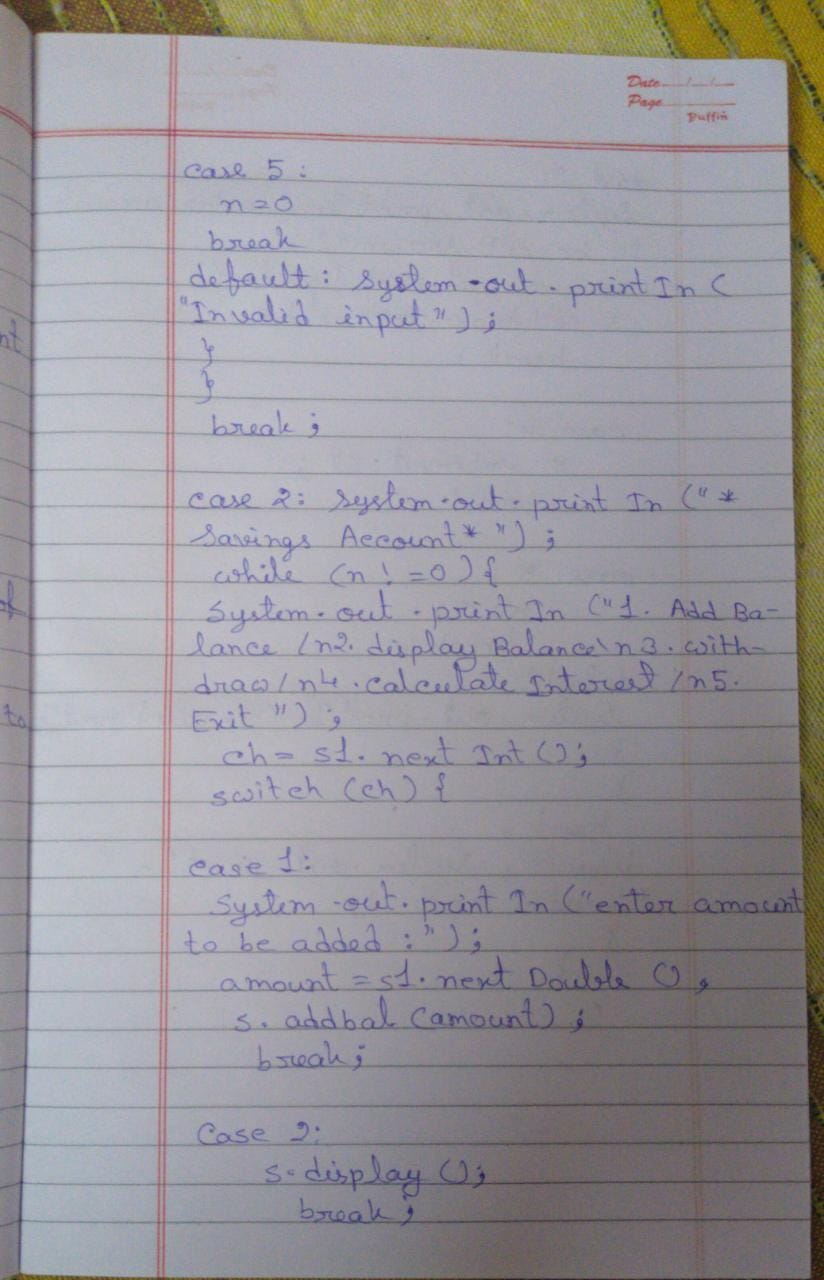
****

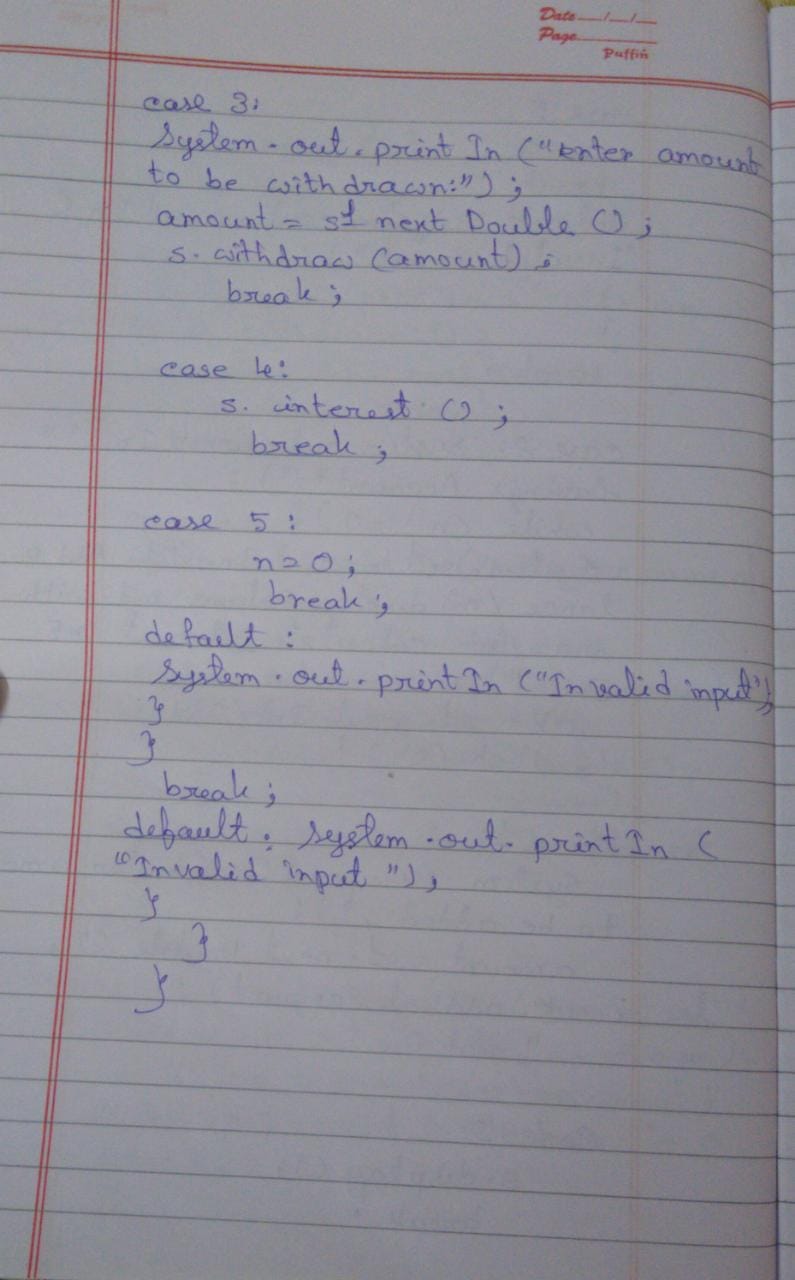
****

****

****

****

****

****

**SOURCE CODE:**

import java.util.Scanner;

abstract class Account{

String c\_name, acc\_type;

int acc\_num;

double balance;

int minbalance = 2000;

Account(String c\_name, int acc\_num, double balance){

this.c\_name = c\_name;

this.acc\_num = acc\_num;

this.balance = balance;

this.acc\_type = acc\_type;

}

abstract void addbal(double amount);

abstract void display();

abstract void withdraw(double amount);

}

class curr\_acct extends Account{

curr\_acct(String c\_name, int acc\_num, double balance){

super(c\_name, acc\_num, balance);

System.out.println("Details ot the customer:");

System.out.println("Customer name: " +c\_name +"\tAccount number: " +acc\_num+ "\tBalance: " +balance+ "Account type: current");

}

void addbal(double amount){

this.balance += amount;

}

void display(){

System.out.println("The balance is:" +this.balance);

}

void withdraw(double amount){

if(this.balance<amount){

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

System.out.println("Yor balance is: " +this.balance);

return;

}

this.balance = this.balance - amount;

if(this.balance<minbalance){

this.balance = this.balance - this.balance\*0.5;

System.out.println("A penalty of Rs. " +this.balance\*0.5 + "has been charged as minimum balance is not satisfied");

System.out.println("Updated Balance: " +this.balance);

System.out.println("Cannot withdraw");

}

else if(balance> minbalance){

this.balance = this.balance - amount;

System.out.println("Balance is: " +this.balance);

}

}

}

class sav\_acct extends Account{

sav\_acct(String c\_name, int acc\_num, double balance){

super(c\_name, acc\_num, balance);

System.out.println("Customer name: " +c\_name +"\tAccount number: " +acc\_num+ "\tBalance: " +balance+ "Account type: savings");

}

void addbal(double amount){

this.balance += amount;

}

void display(){

System.out.println("The balance is:" +this.balance);

}

void withdraw(double amount){

if(this.balance<amount){

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

System.out.println("Yor balance is: " +this.balance);

}

this.balance = this.balance - amount;

if(this.balance<minbalance){

this.balance = this.balance - this.balance\*0.5;

System.out.println("A penalty of Rs. " +this.balance\*0.5 + "has been charged as minimum balance is not satisfied");

System.out.println("Updated Balance: " +this.balance);

System.out.println("Cannot withdraw");

}

else if(balance> minbalance){

this.balance = this.balance - amount;

System.out.println("Balance is: " +this.balance);

}

}

void interest(){

int time = 3, n=1;

System.out.println("Rate of intrest is 0.2");

this.balance = this.balance\*Math.pow(1+(0.2)/n, (n\*time));

System.out.println("Amount after adding interest is "+this.balance);

}

}

class Bank{

public static void main(String args[]){

int choice,ch,n=1;

double amount;

Scanner s1 = new Scanner(System.in);

curr\_acct c = new curr\_acct("Ram", 12345, 50000);

sav\_acct s = new sav\_acct("Ram", 12345, 50000);

System.out.println("Press 1.For Current account\nPress 2.For Savings account");

choice = s1.nextInt();

switch(choice){

case 1: System.out.println("\*Current Account\*");

while(n!=0){

System.out.println("1.AddBalance\n2.displayBalance\n3.withdraw\n4.checkbook\n5.Exit");

ch = s1.nextInt();

String reciever;

double recamount;

switch(ch){

case 1:

System.out.println("enter amount to be added:");

amount = s1.nextDouble();

c.addbal(amount);

break;

case 2:

c.display();

break;

case 3:

System.out.println("enter amount to be withdrawn:");

amount = s1.nextDouble();

c.withdraw(amount);

break;

case 4:

System.out.println("Enter the name of the reciever:");

reciever = s1.next();

System.out.println("Enter amount to be debited to reciever:");

recamount = s1.nextDouble();

if(recamount>c.balance){

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

}

else{

System.out.println("Amount of " +recamount+ " sent to " +reciever);

c.balance = c.balance - recamount;

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

s.balance = c.balance;

}

case 5:

n=0;

break;

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

}

}

break;

case 2: System.out.println("\*Savings Account\*");

while(n!=0){

System.out.println("1.AddBalance\n2.displayBalance\n3.withdraw\n4.CalculateInterest\n5.Exit");

ch = s1.nextInt();

switch(ch){

case 1:

System.out.println("enter amount to be added:");

amount = s1.nextDouble();

s.addbal(amount);

break;

case 2:

s.display();

break;

case 3:

System.out.println("enter amount to be withdrawn:");

amount = s1.nextDouble();

s.withdraw(amount);

break;

case 4:

s.interest();

break;

case 5:

n=0;

break;

default:

System.out.println("Invalid input");

}

}

break;

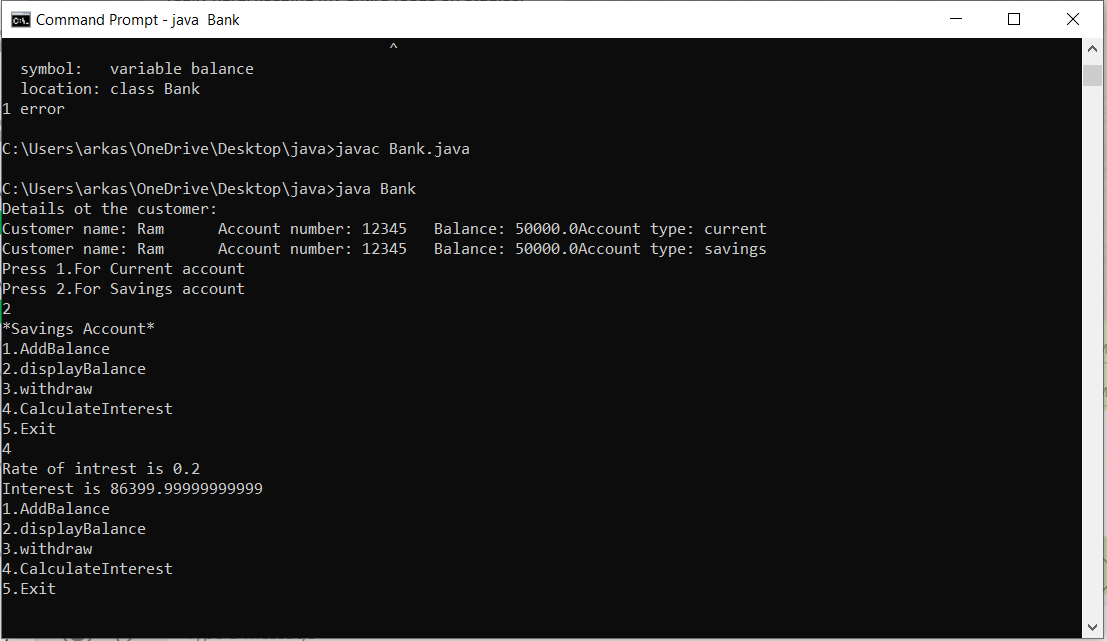
default: System.out.println("Invalid input");

}

}

}

**OUTPUT:**

****