**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Belgaum -590014, Karnataka.**

****

**LAB REPORT**

**on**

**OBJECT ORIENTED JAVA PROGRAMMING (21CS3PCOOJ)**

***Submitted by***

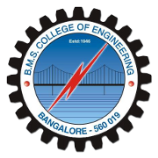
**R Likith (1BM21CS151)**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING**

**(Autonomous Institution under VTU)**

**BENGALURU-560019**

**October-2022 to Feb-2023**

**B. M. S. College of Engineering,**

**Bull Temple Road, Bangalore 560019**

(Affiliated To Visvesvaraya Technological University, Belgaum)

**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “Object Oriented Java Programming (21CS3PCDOOJ)” carried out by **R Likith (1BM21CS151),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of Object Oriented Java Programming (21CS3PCOOJ) work prescribed for the said degree.

**Basavaraj Jakkali** **Dr. Jyothi S NayaK**

Associate Professor Professor and Head Department of CSE Department of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

PROGRAM 1

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.

CODE

import java.util.\*;

import java.lang.Math;

class prog1{

public static void main (String[] args)

{

Scanner sc = new Scanner (System.in);

System.out.println("Enter the values of a,b,c");

float a = sc.nextFloat();

float b = sc.nextFloat();

float c = sc.nextFloat();

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

double r1, r2;

if (a==0) {

System.out.println("Not a quadratic equation"); }

else if (d==0){

System.out.println("Roots are equal.");

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

System.out.println ("Root 1 and Root 2 are: "+r1); }

else if (d>0){

System.out.println ("Roots are real and distinct.");

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

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

System.out.println ("Root 1: "+r1);

System.out.println ("Root 2: " +r2); }

else {

System.out.println ("Roots are imaginary");

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

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

System.out.println ("Root 1: " +r1 + "+i" +r2);

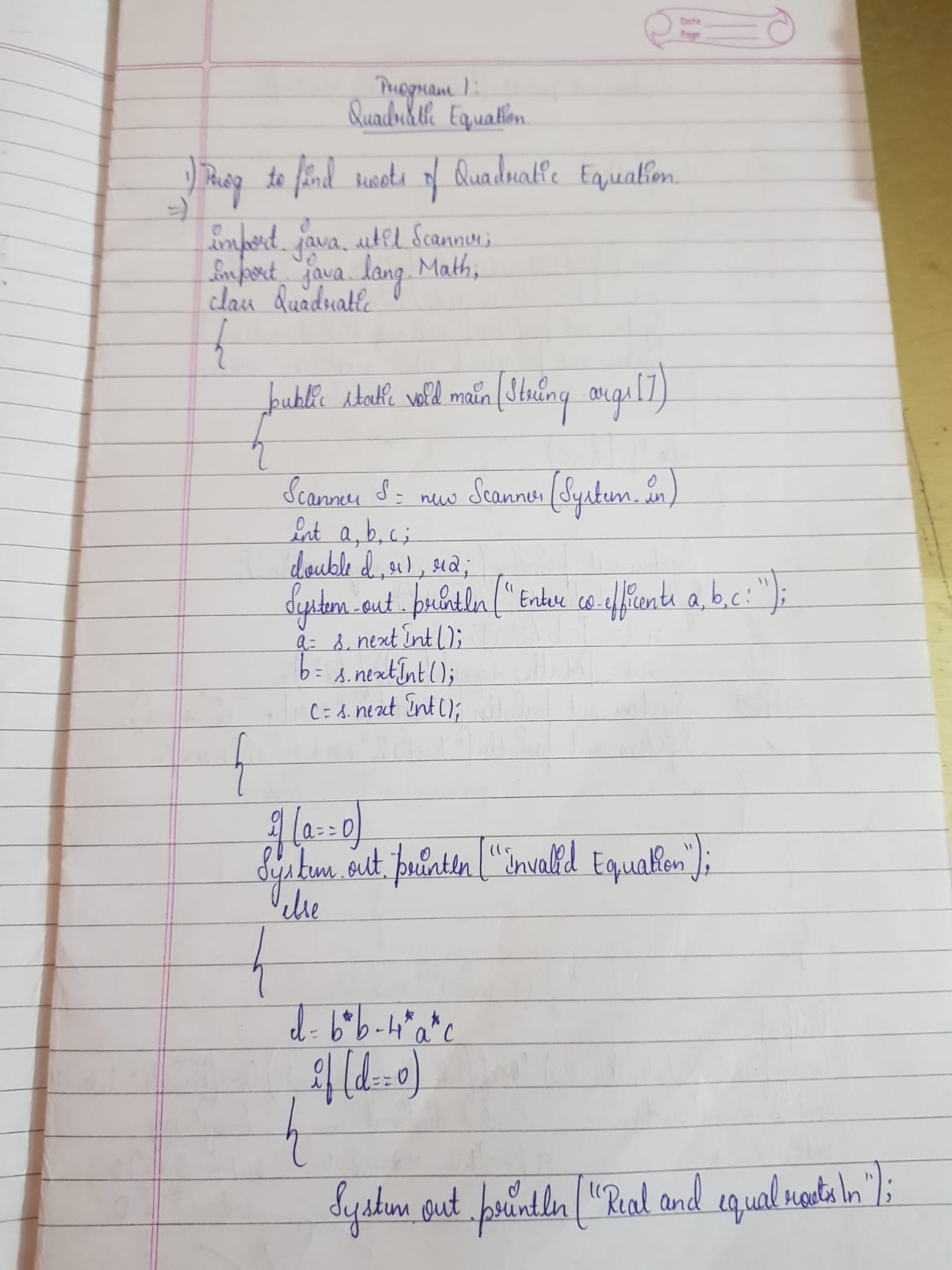
System.out.println ("Root 2: " +r1 + "-i" +r2);

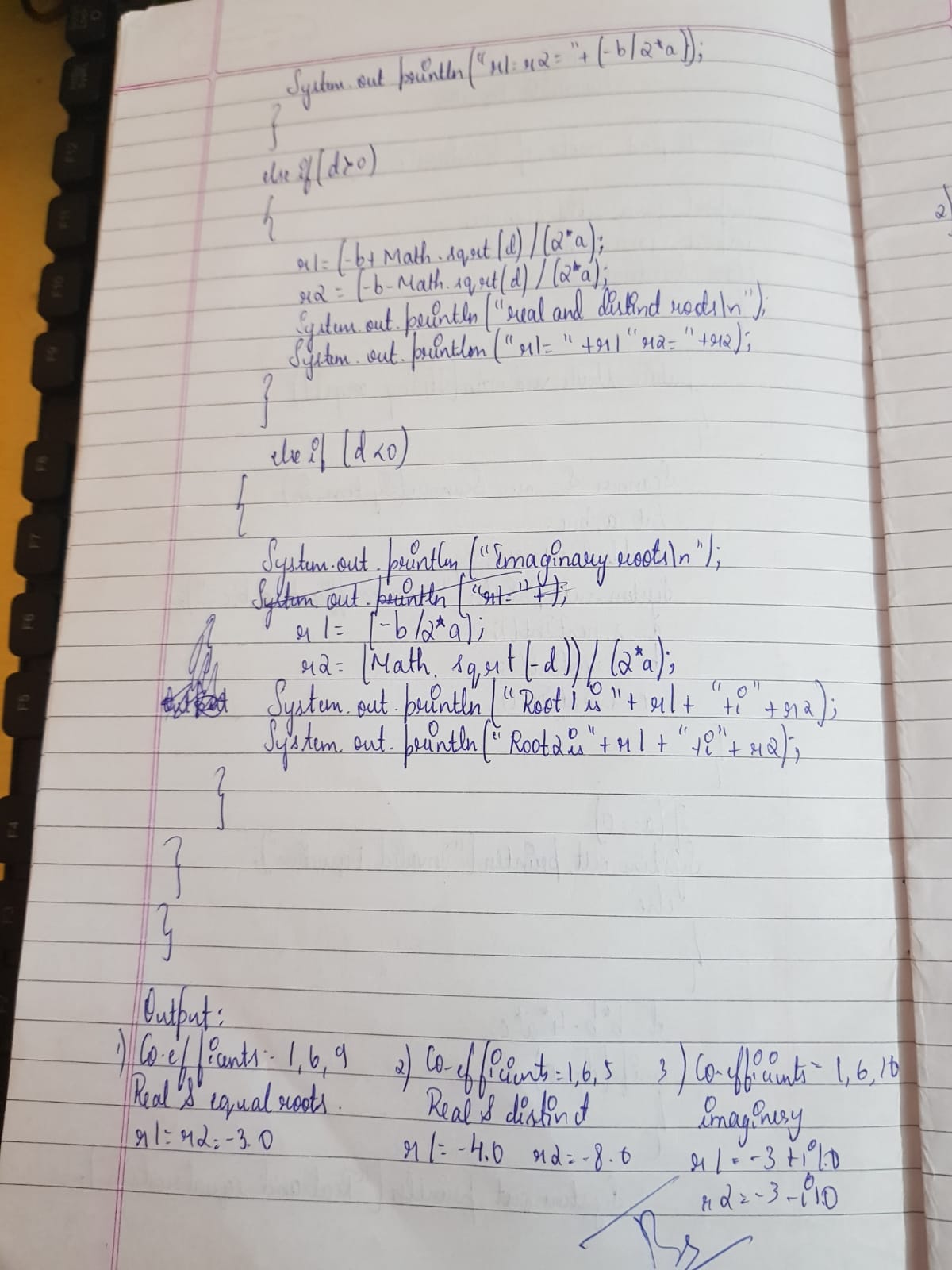
}

}

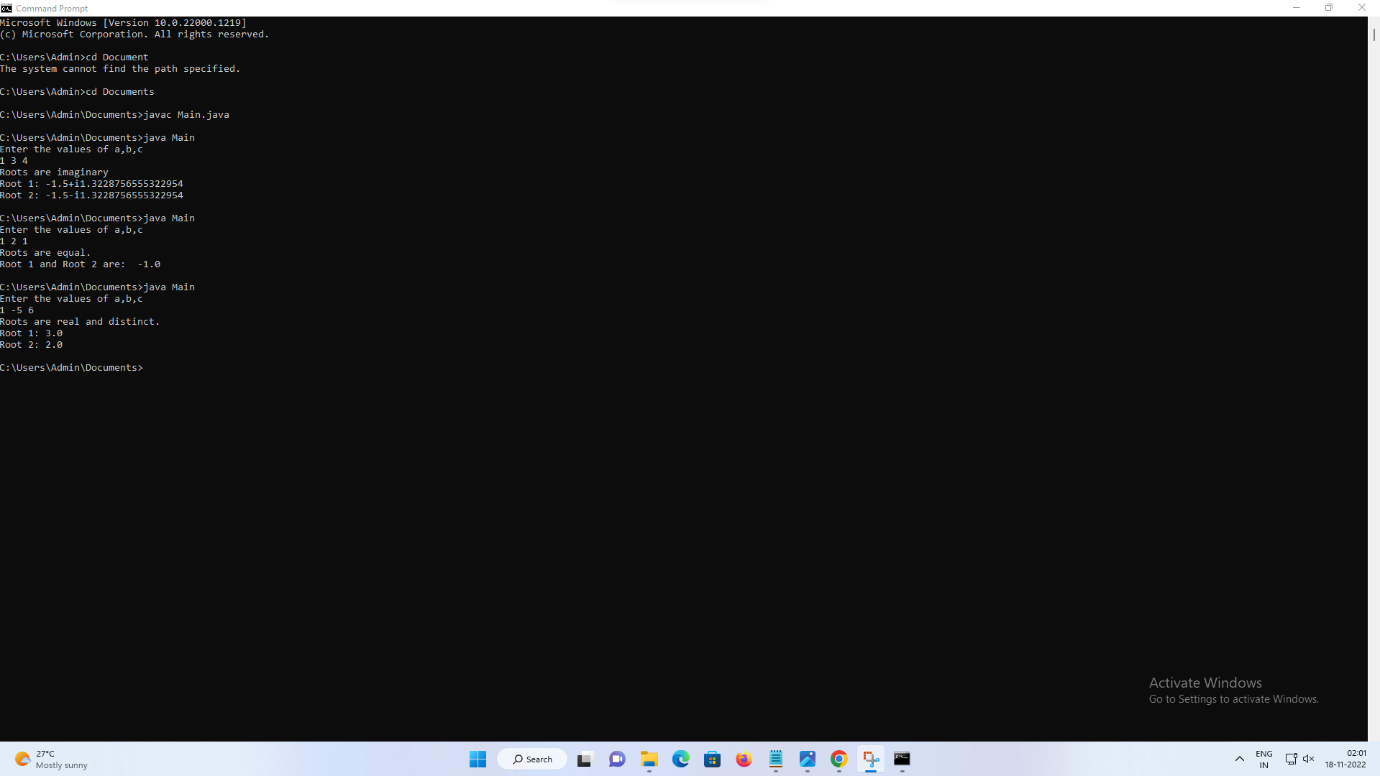
}

OBSERVATION





OUTPUT



PROGRAM 2

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.

CODE

import java.util.\*;

class Student{

String USN;

String name;

int sub;

int[] marks = new int[10];

int[] credits = new int[10];

int gradepoints[] = new int[10];

double nume =0;

double denom=0;

double SGPA;

void accept()

{

Scanner sc = new Scanner (System.in);

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

USN = sc.nextLine();

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

name = sc.nextLine();

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

sub = sc.nextInt();

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

{

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

marks[i] = sc.nextInt();

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

credits[i] = sc.nextInt();

denom += credits[i];

}

}

void calculate()

{

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

{

if(marks[i]>=90)

gradepoints[i]=10;

else if (marks[i]>= 80 && marks[i]<90)

gradepoints[i]=9;

else if (marks[i]>= 70 && marks[i]<80)

gradepoints[i]=8;

else if (marks[i]>= 60 && marks[i]<70)

gradepoints[i]=7;

else if (marks[i]>= 50 && marks[i]<60)

gradepoints[i]=6;

else if (marks[i]>= 40 && marks[i]<50)

gradepoints[i]=5;

else

gradepoints[i]=0;

nume += credits[i]\*gradepoints[i]; }

SGPA = nume/denom;

}

void display()

{

System.out.println("The student details are:");

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

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

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

}

}

class prog2{

public static void main (String args[]) {

Student s = new Student();

s.accept();

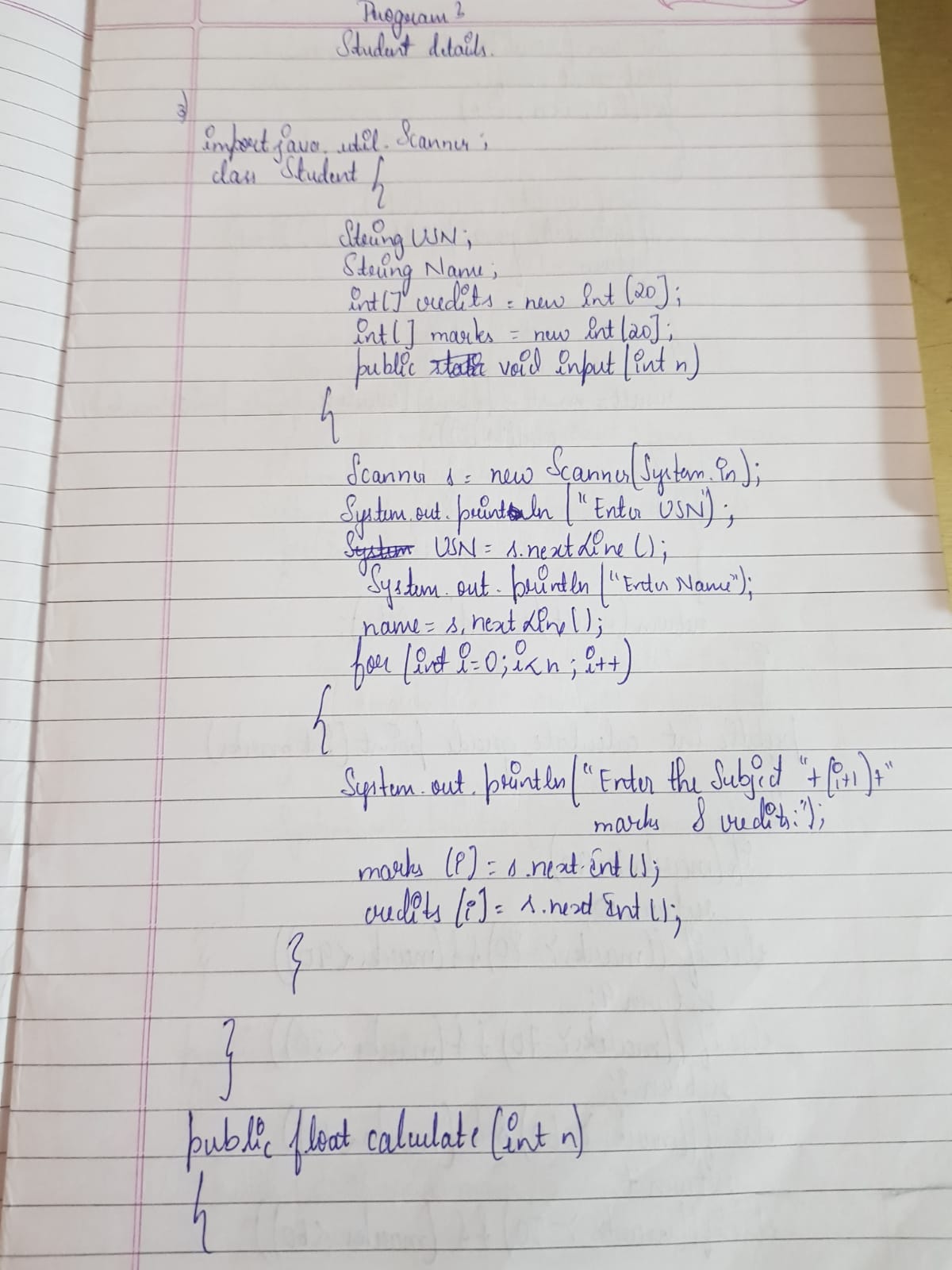
s.calculate();

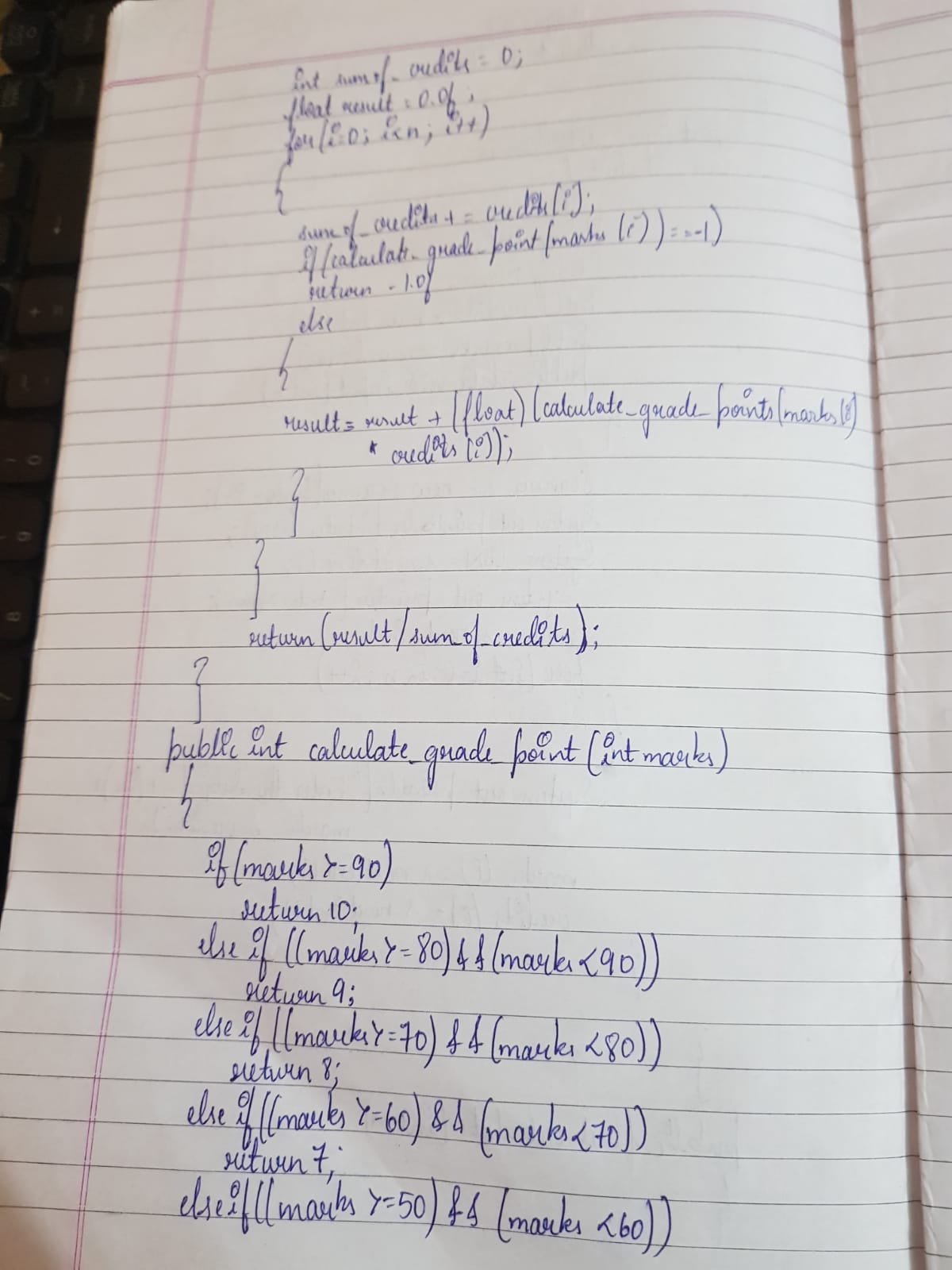
s.display();

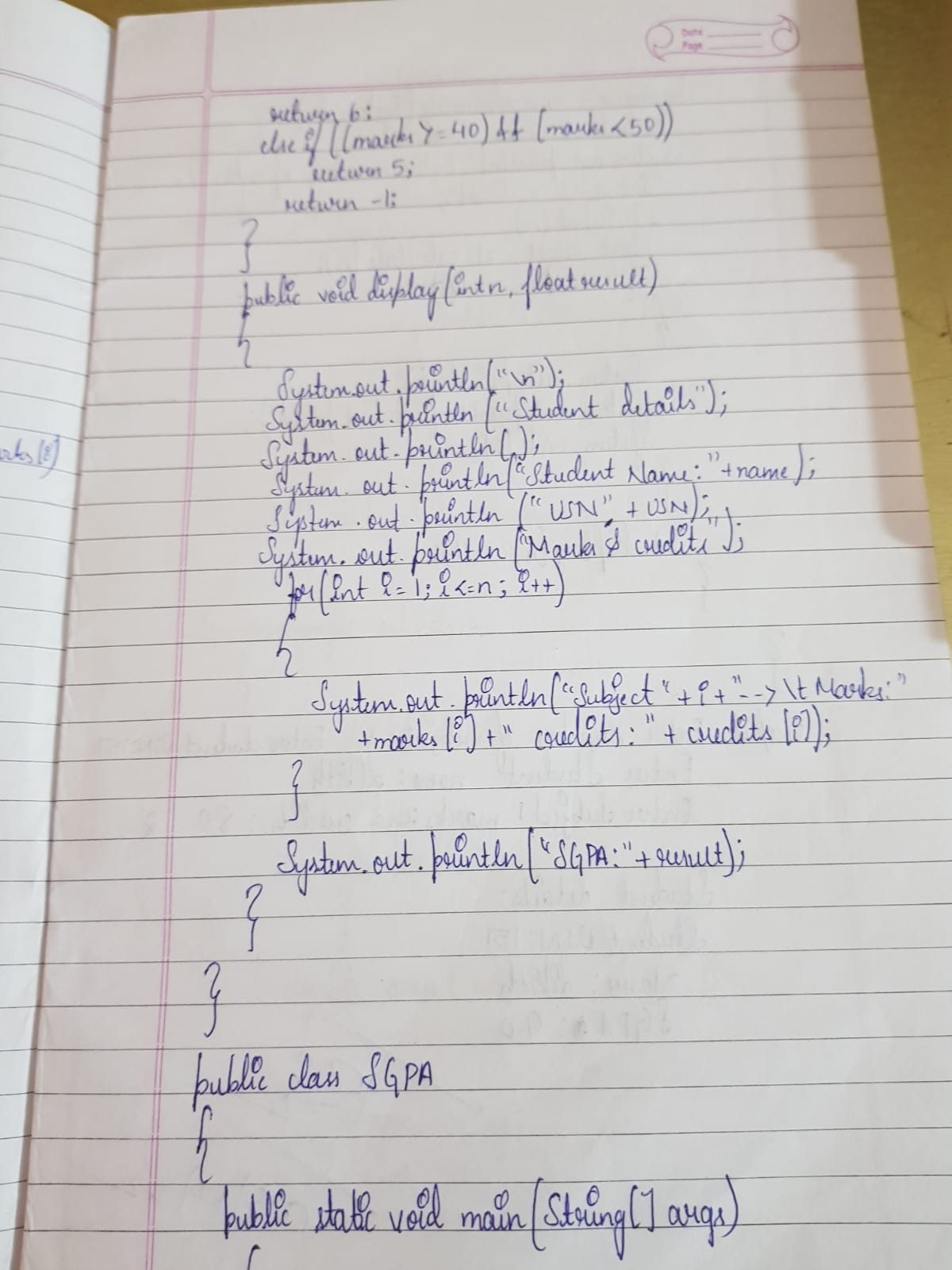
}

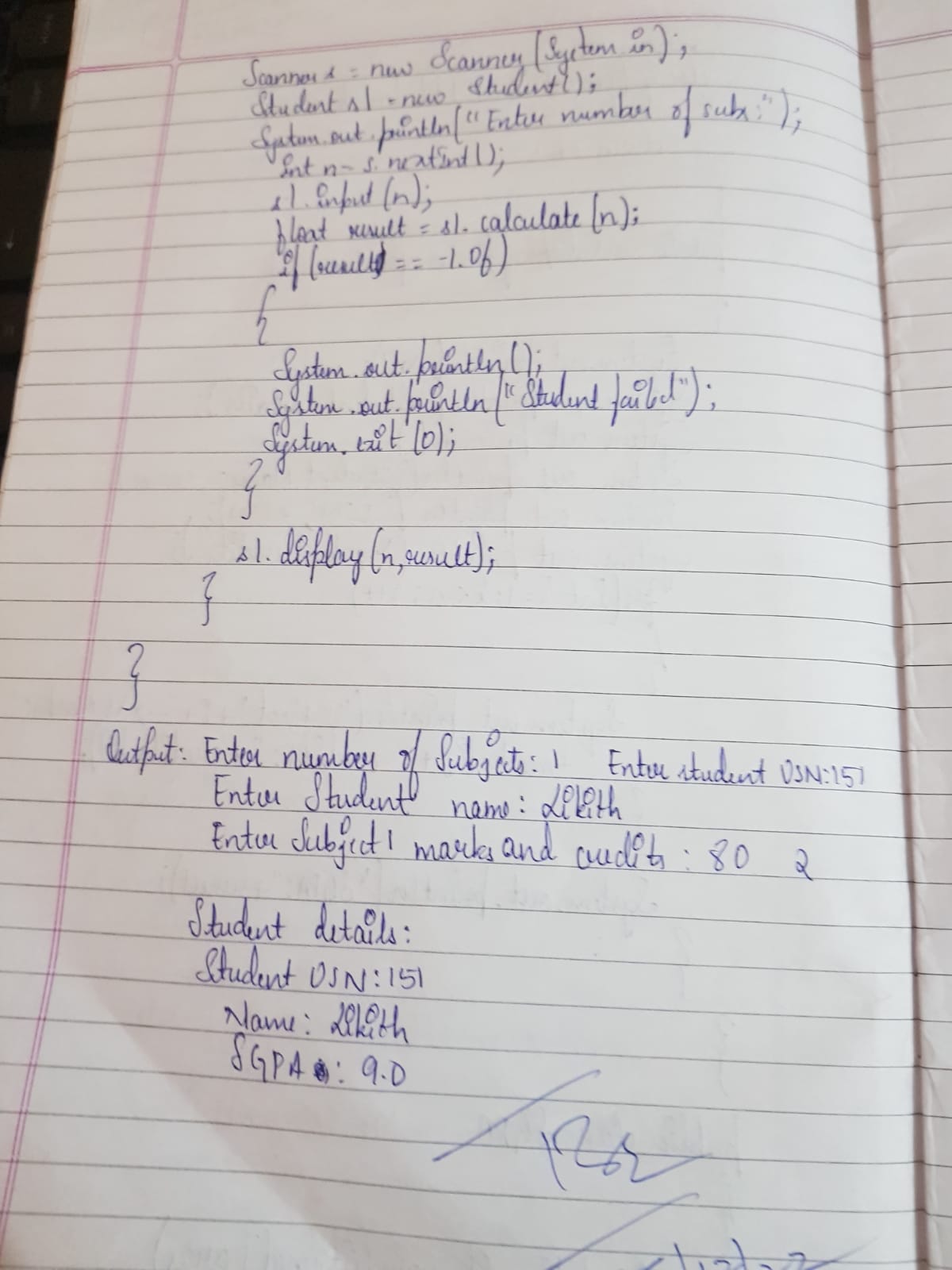
}

OBSERVATION

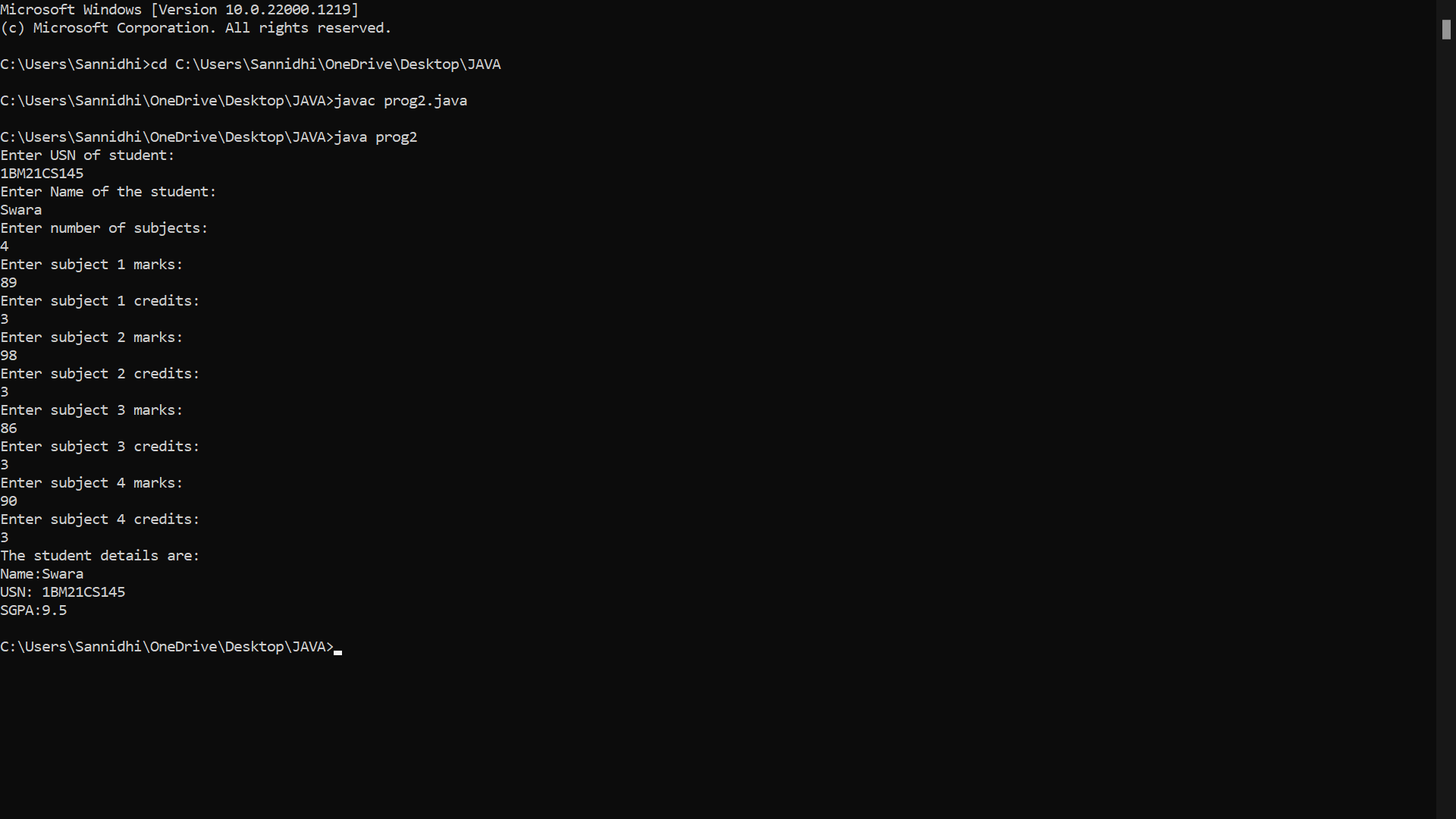








OUTPUT



PROGRAM 3

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.

CODE

import java.io.\*;

import java.util.\*;

class Book {

String title, author;

double price;

int numPages;

Book() {

title="Default";

author="Default";

price=0.0;

numPages=0; }

void setTitle(String t) {

title=t; }

void setAuthor(String a) {

author=a;}

void setPrice(double p) {

price=p;}

void setPages(int np) {

numPages=np;}

public String toString() {

return title+"\t"+author+"\t"+price+"\t"+numPages+"\n";}}

class prog3 {

public static void main(String args[]) {

String t, a;

double p;

int np,n;

Scanner sc = new Scanner(System.in);

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

n = sc.nextInt();

Book b[]= new Book[n];

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

System.out.println("Enter the Title of the Books");

t= sc.next();

System.out.println("Enter the Author of the Books");

a= sc.next();

System.out.println("Enter the Price of the Books"); p= sc.nextDouble();

System.out.println("Enter the Number of pages of the Books");

np= sc.nextInt();

b[i] = new Book();

b[i].setTitle(t);

b[i].setAuthor(a);

b[i].setPrice(p);

b[i].setPages(np);

}

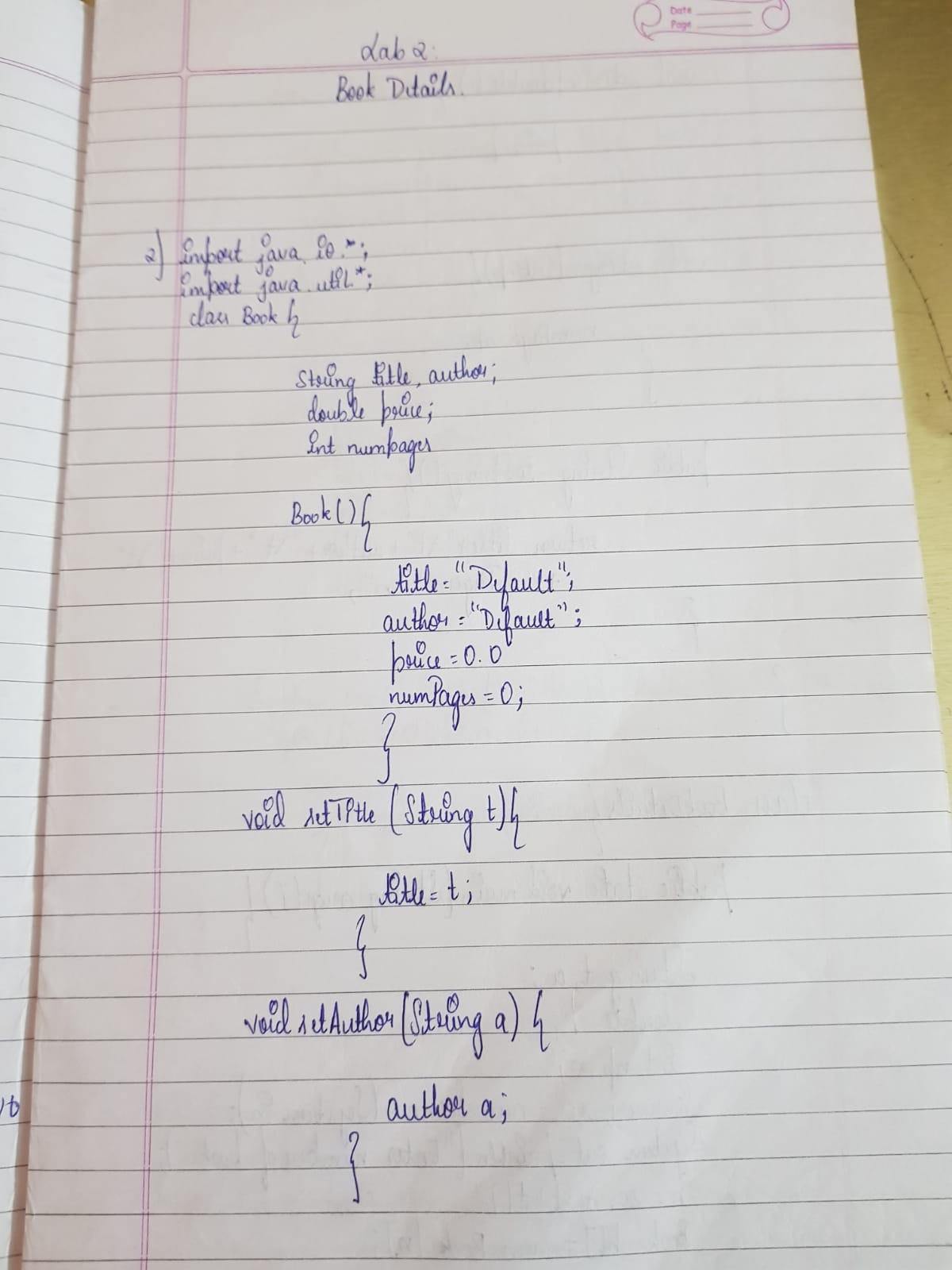
System.out.println("Title \t Author \t Price \t Pages\n");

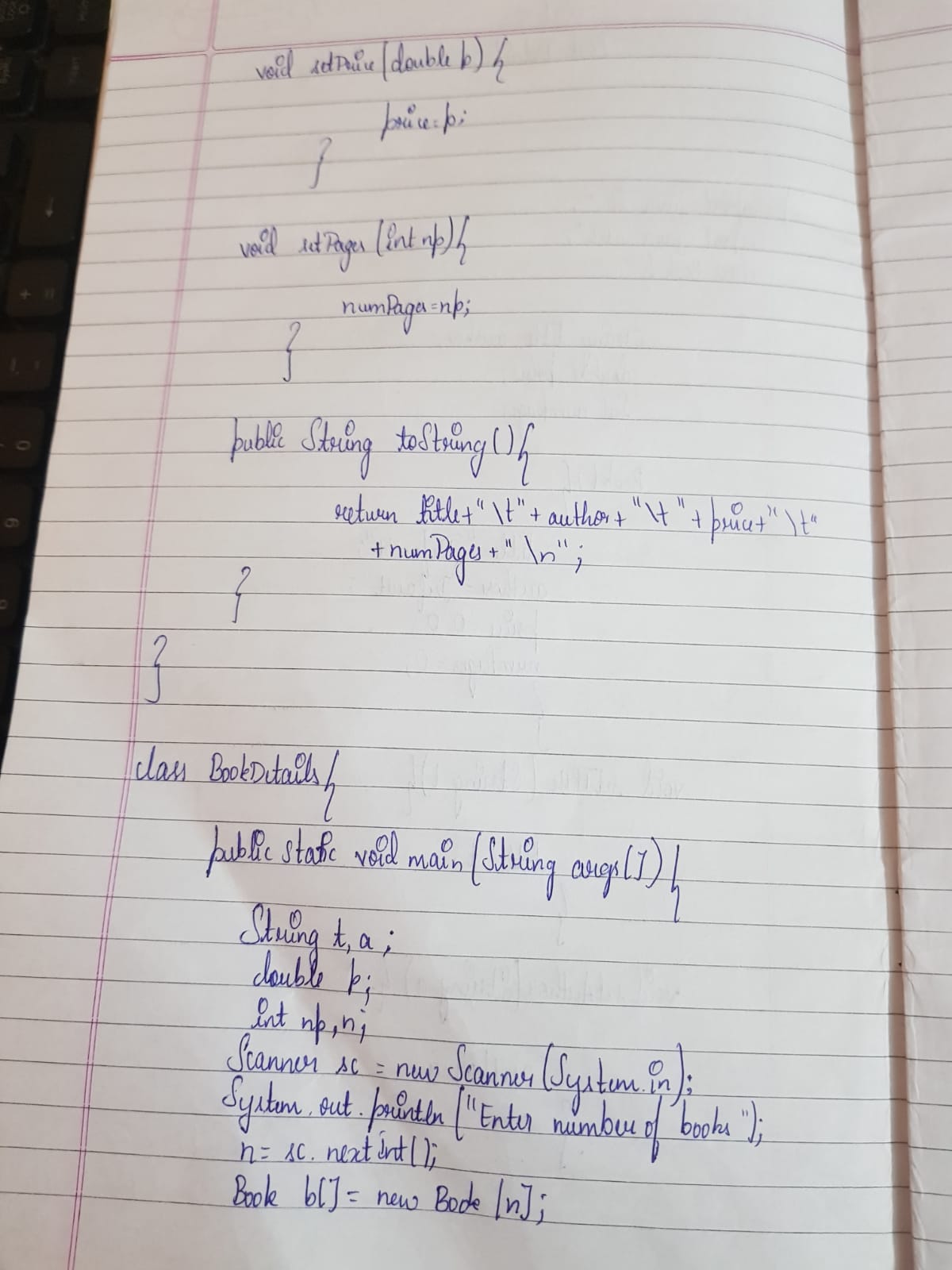
for(int i=0; i<n;i++) {

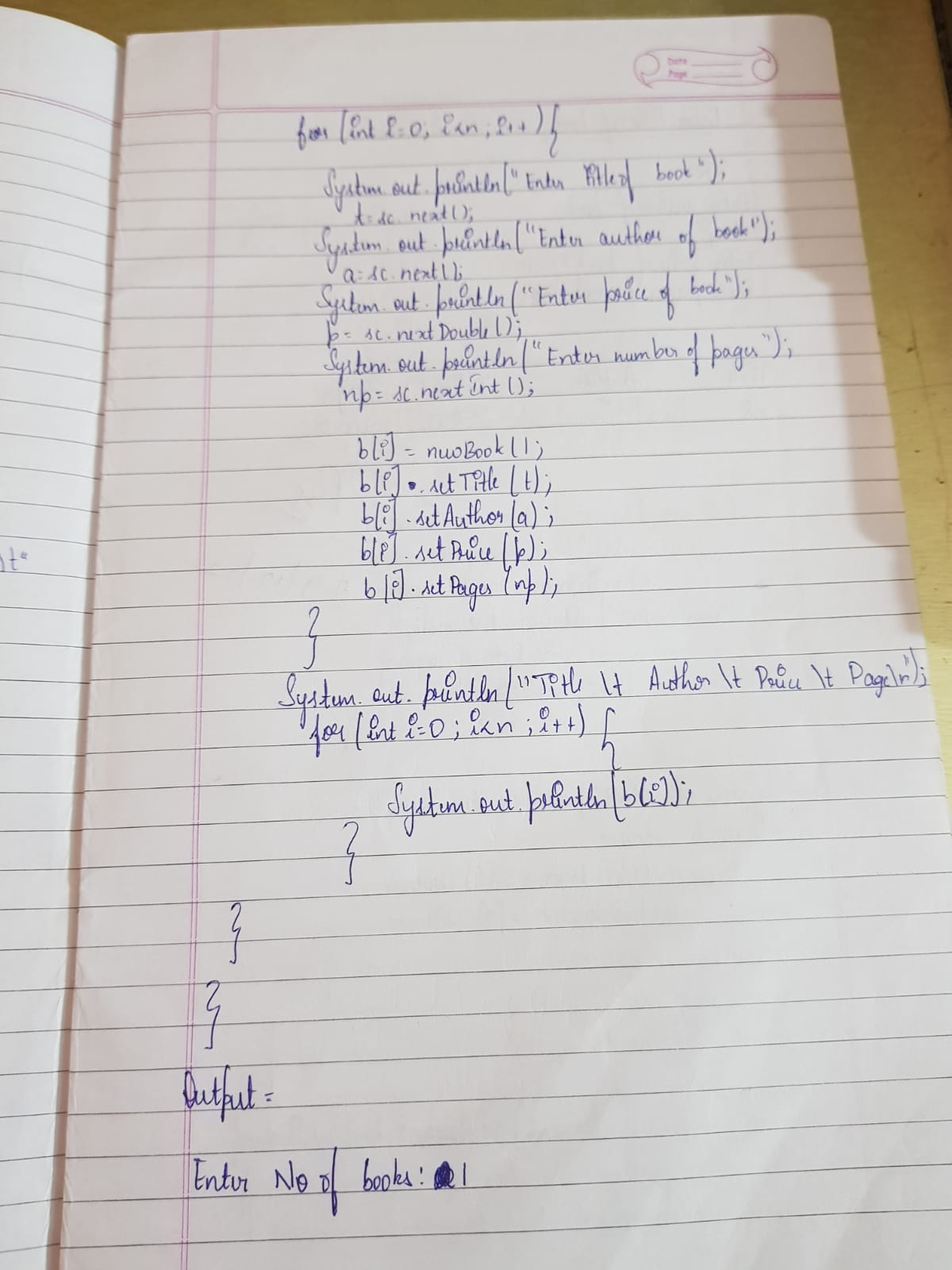
System.out.println(b[i]); }

}

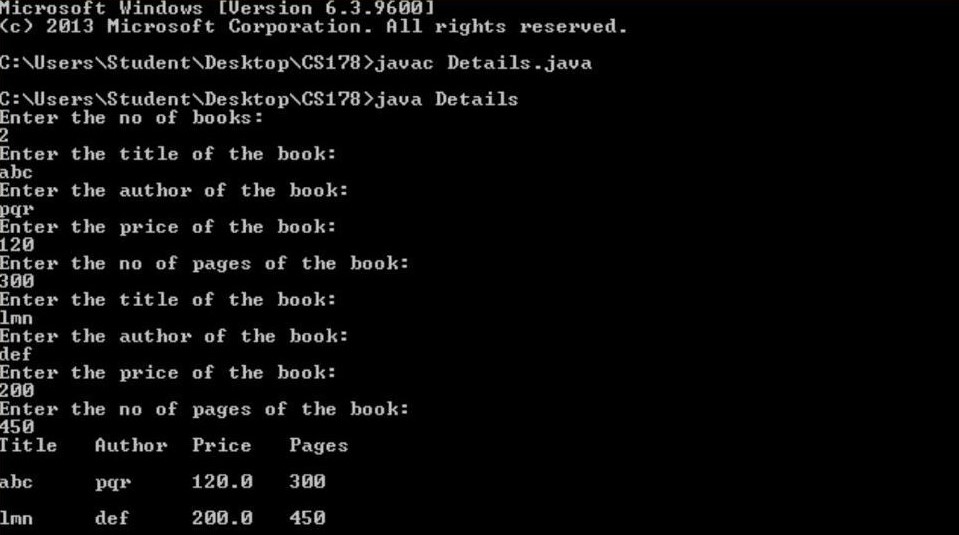
OBSERVATION







OUTPUT



PROGRAM 4

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.

CODE

import java.util.Scanner;

abstract class Shape

{

int a,b;

abstract void printArea();

}

class Rectangle extends Shape

{

void printArea()

{

int area;

Scanner s = new Scanner(System.in);

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

a = s.nextInt();

b = s.nextInt();

area = a\*b;

System.out.println("\nArea of Rectangle: "+area+"\n");

}

}

class Square extends Shape

{

void printArea()

{

int area;

Scanner s = new Scanner(System.in);

System.out.println("Enter the side of square: ");

a = s.nextInt();

area = a\*a;

System.out.println("\nArea of square: "+area+"\n");

}

}

class Circle extends Shape{

void printArea()

{

double area;

Scanner s = new Scanner(System.in);

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

a = s.nextInt();

area = Math.PI\*a;

System.out.println("Area of Circle: "+area+"\n");

}

}

public class prog4

{

public static void main(String[] args)

{

int choice;

Scanner s = new Scanner(System.in);

do

{

System.out.println("1. Calculate Area of Rectangle\n2. Calculate Area of Square\n3. Calculate Area of " +

"Circle\n4. Exit the Program\n\nEnter the choice: ");

choice = s.nextInt();

switch(choice)

{

case 1: Rectangle r = new Rectangle();

r.printArea();

break;

case 2: Square sq = new Square();

sq.printArea();

break;

case 3: Circle c = new Circle();

c.printArea();

break;

case 4: System.out.println("Exiting the program!");

System.exit(0);

break;

default: System.out.println("\nInvalid Choice!\n");

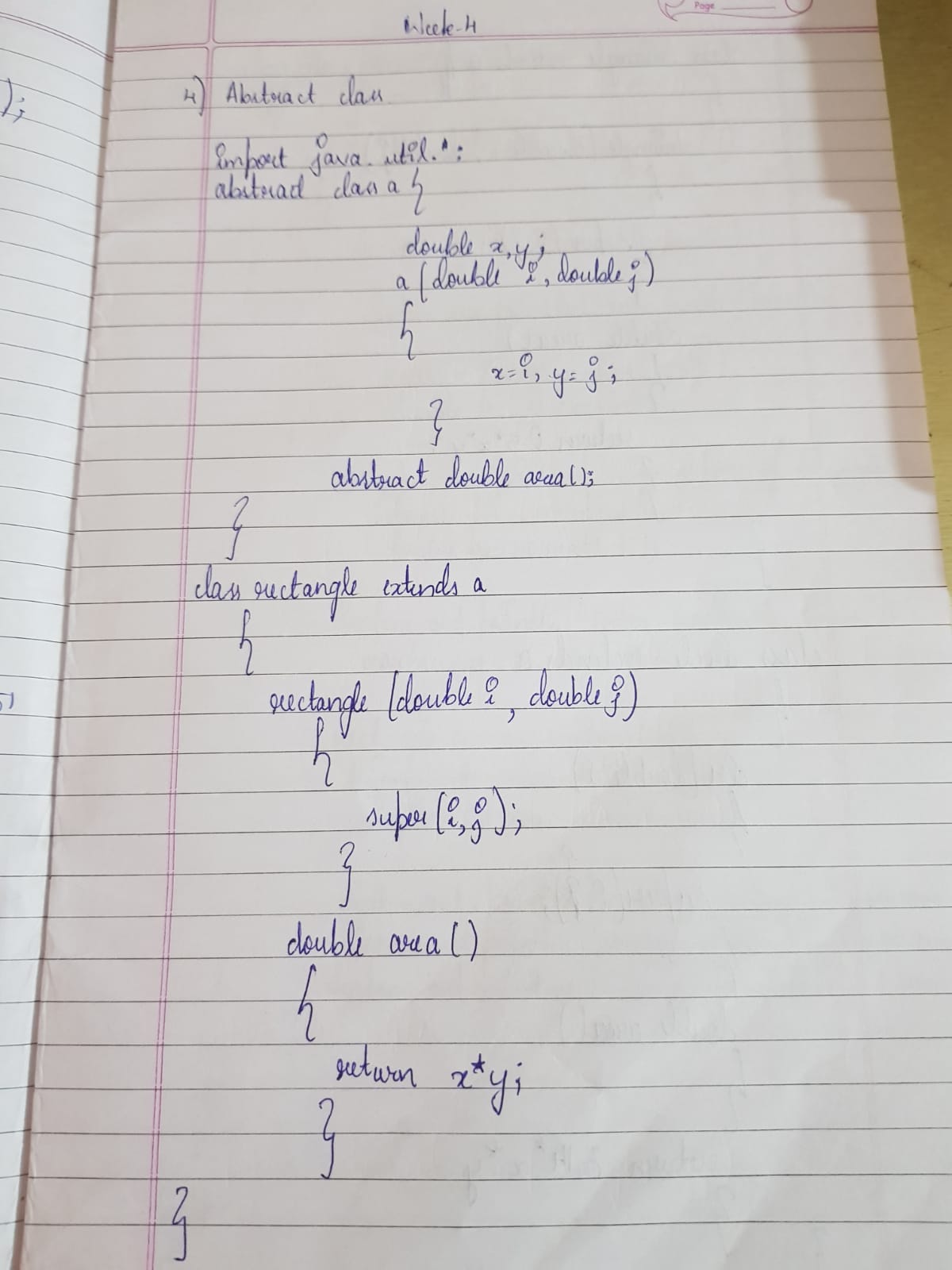
}

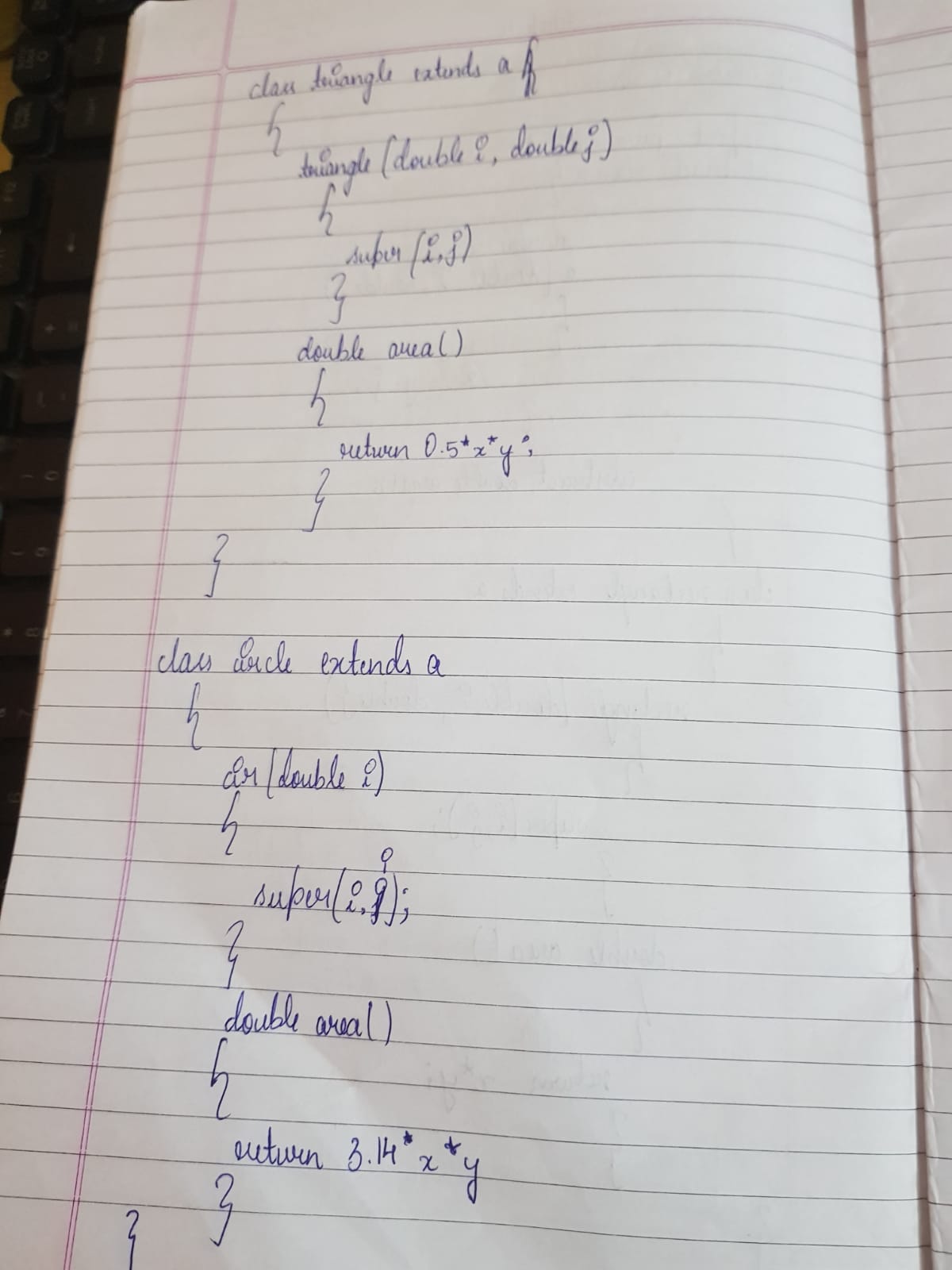
}while(true);

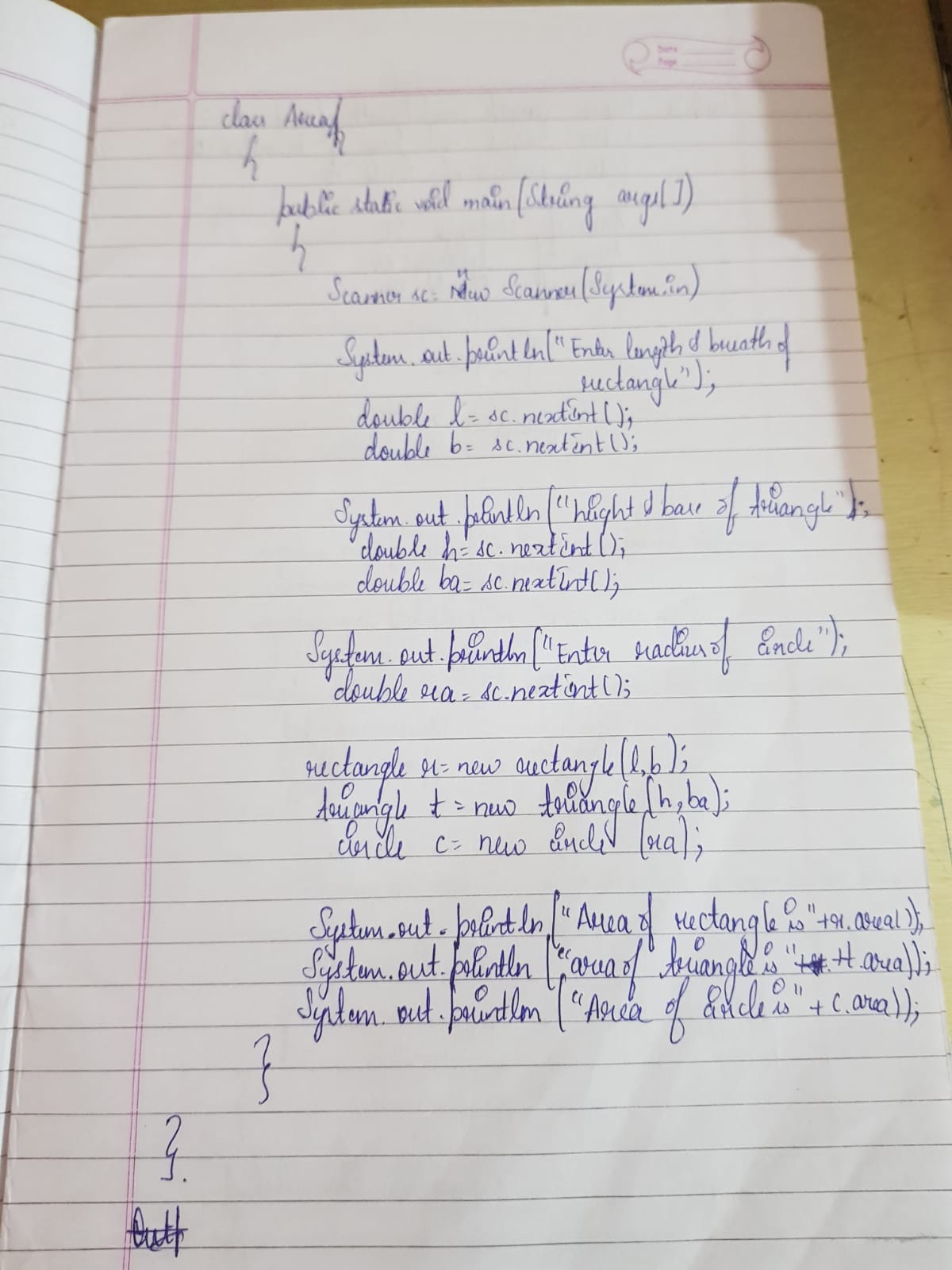
}

}

OBSERVATION







OUTPUT



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.

CODE

import java.util.Scanner;

class Account

{

String customer\_name;

long acc\_no;

float bal;

Scanner s = new Scanner(System.in);

public void input()

{

System.out.print("\nEnter the Customer Name: ");

customer\_name = s.nextLine();

System.out.print("\nEnter the Account Number: ");

acc\_no = s.nextLong();

System.out.print("\nEnter the Starting Amount (Minimum Amount = 5000): ");

bal = s.nextFloat();

if(bal<5000f){

System.out.println("\nAccount Balance cannot be less than 5000.0 \n");

System.exit(0);

}

}

public void display() {

System.out.println("\nCustomer Name: "+customer\_name);

System.out.println("Account Number: "+acc\_no);

System.out.println("Amount: "+bal);

}

}

class Savings extends Account

{

Scanner s = new Scanner(System.in);

float deposit,withdraw,interest;

public void deposit()

{

System.out.print("\nEnter the amount to be deposited: ");

deposit = s.nextFloat();

bal+=deposit;

System.out.println("\nBalance: "+bal);

}

public void withdraw()

{

System.out.print("\nEnter the amount to be withdrawn: ");

withdraw = s.nextFloat();

if( bal<5000 || bal<withdraw )

{

System.out.println("\nInsufficient Balance");

}

else

{

bal-=withdraw;

System.out.println("\nAmount Withdrawn: "+withdraw+"\nBalance: "+bal);

}

}

public void check\_Bal(){

if(bal<5000)

{

System.out.println("\nInsufficient Balance!!\nBalance: "+bal);

}

else

{

System.out.println("\nBalance: "+bal);

}

}

public void interest()

{

interest=(bal\*6)/100;

bal+=interest;

System.out.println("\nInterest Credited: "+interest+"\nBalance :"+bal) ;

}

}

class Current extends Account

{

float deposit, withdraw, penalty;

public void deposit()

{

System.out.print("\nEnter Amount to be deposited: ");

deposit = s.nextFloat();

bal += deposit;

System.out.println("Balance: " + bal);

}

public void check\_Bal()

{

if (bal < 5000)

{

penalty = (0.1f \* bal);

System.out.println("\nInitial Account Balance: "+bal);

bal = bal-penalty;

System.out.println("\nLow balance!\nPenalty Amount: " + penalty + "\nAccount balance: " + bal);

}

else

{

System.out.println("\n Balance: " + bal);

}

}

public void withdraw()

{

System.out.print("\nEnter Amount to withdraw: ");

withdraw = s.nextFloat();

if(bal<5000)

{

check\_Bal();

}

else if(bal<withdraw)

{

System.out.println("\nInsufficient Balance!!\nBalance: "+bal);

}

else

{

bal-=withdraw;

System.out.println("\nAmount Withdrawn: "+withdraw+"\nBalance: "+bal);

}

}

public void chequebook()

{

System.out.println("\nCheque Book has been Issued!");

}

}

public class prog5

{

public static void main(String[] args)

{

Scanner s = new Scanner(System.in);

String ch;

int n;

Current c = new Current();

Savings sa = new Savings();

System.out.print("\nEnter the Account Type (S for Savings , C for Current) : ");

ch = s.next();

switch(ch.toLowerCase())

{

case "s" : sa.input();

do

{

System.out.print("\n1. Deposit \n2. Withdrawal \n3. Check Balance \n4. Check Interest\n5. Show Account Details \n6. Exit Transaction\n\nEnter your choice: ");

n = s.nextInt();

switch(n)

{ case 1 : sa.deposit();

break;

case 2 : sa.withdraw();

break;

case 3 : sa.check\_Bal();

break;

case 4 : sa.interest();

break;

case 5 : sa.display();

break;

case 6 : System.out.println("\nExiting Transaction!");

System.exit(0);

break;

default : System.out.println("\nInvalid Operation");

}

}while(true);

case "c" : c.input();

do {

System.out.print("\n1. Deposit \n2. Withdrawal \n3. Check Balance \n4. Issue Cheque Book "\n5. Show Account Details \n6. Exit Transaction\n\nEnter your choice: ");

n = s.nextInt();

switch (n) {

case 1:

c.deposit();

break;

case 2:

c.withdraw();

break;

case 3:

c.check\_Bal();

break;

case 4:

c.chequebook();

break;

case 5:

c.display();

break;

case 6:

System.out.println("\nExiting Transaction!");

System.exit(0);

break;

default:

System.out.println("\nInvalid Operation");

}

}while(true);

default : System.out.println("\nInvalid Choice");

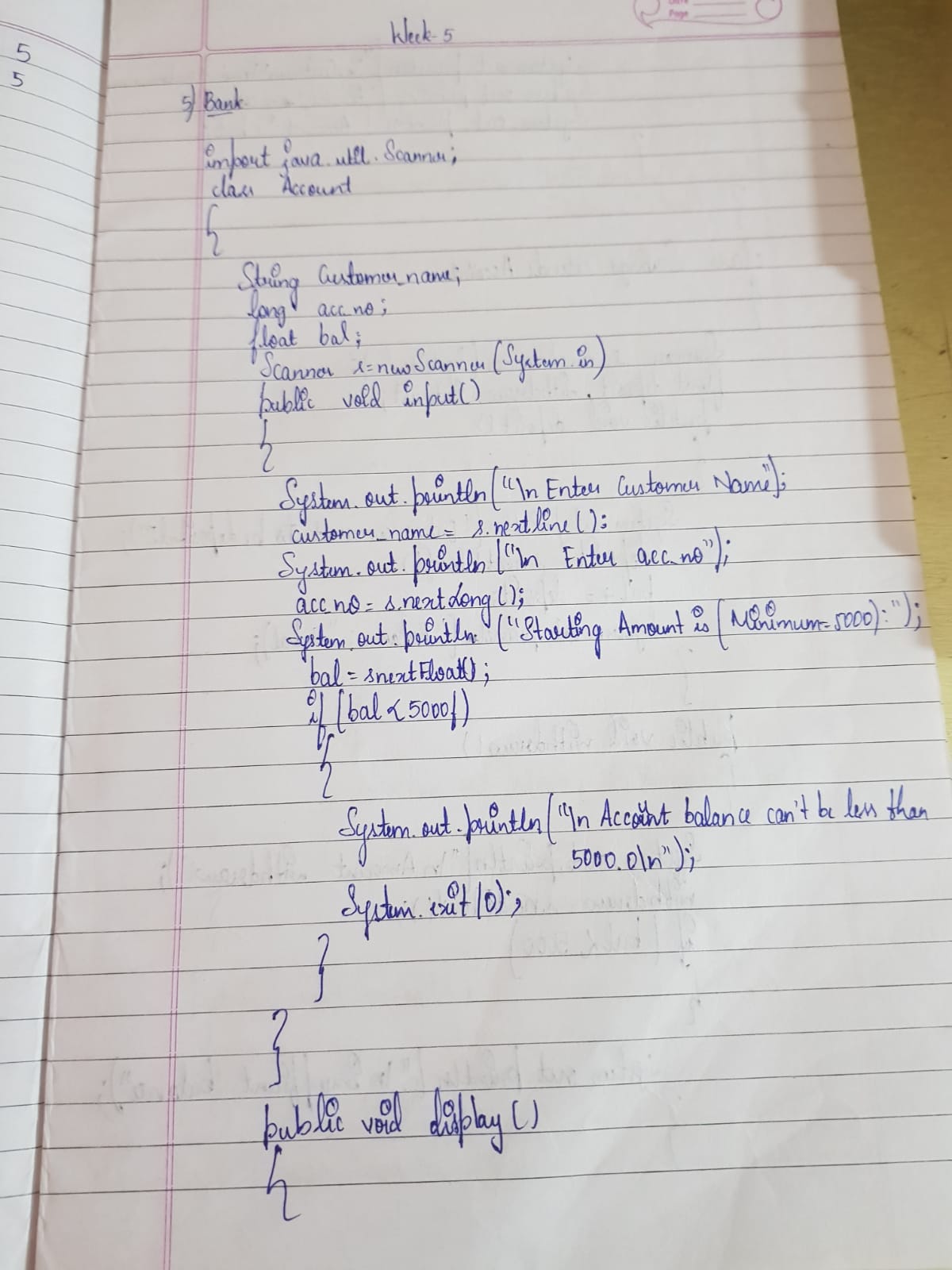
break;

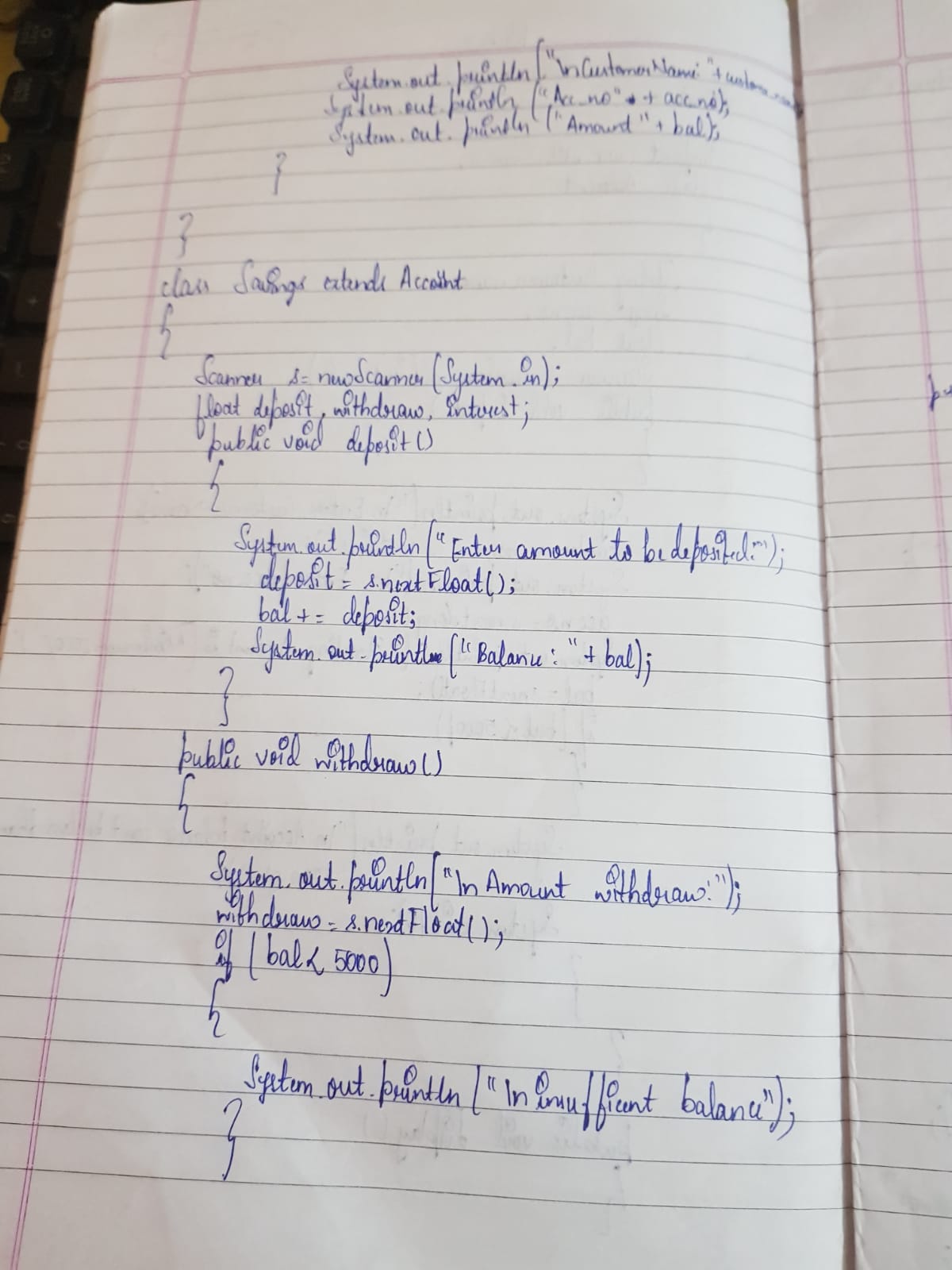
}

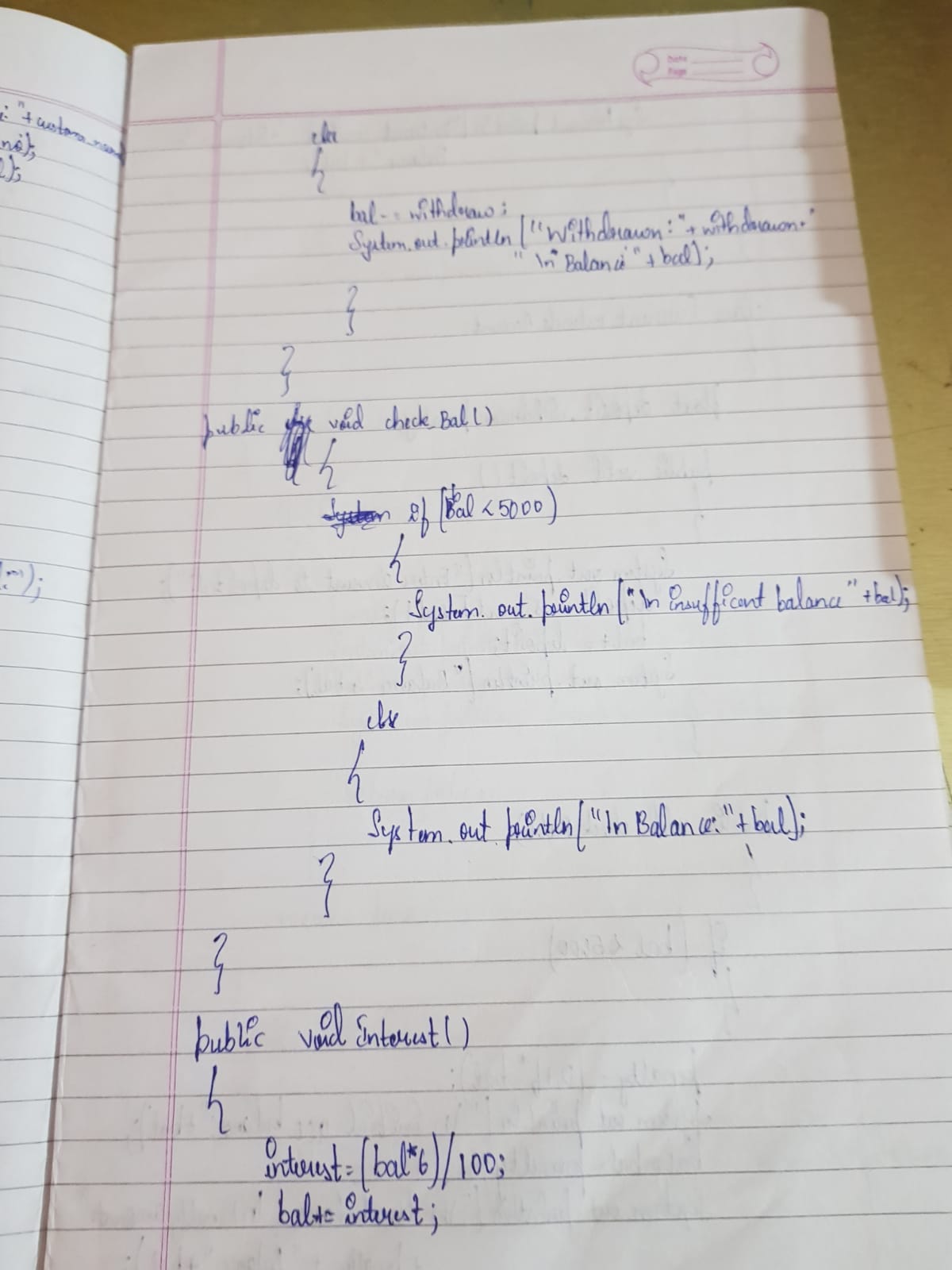
}

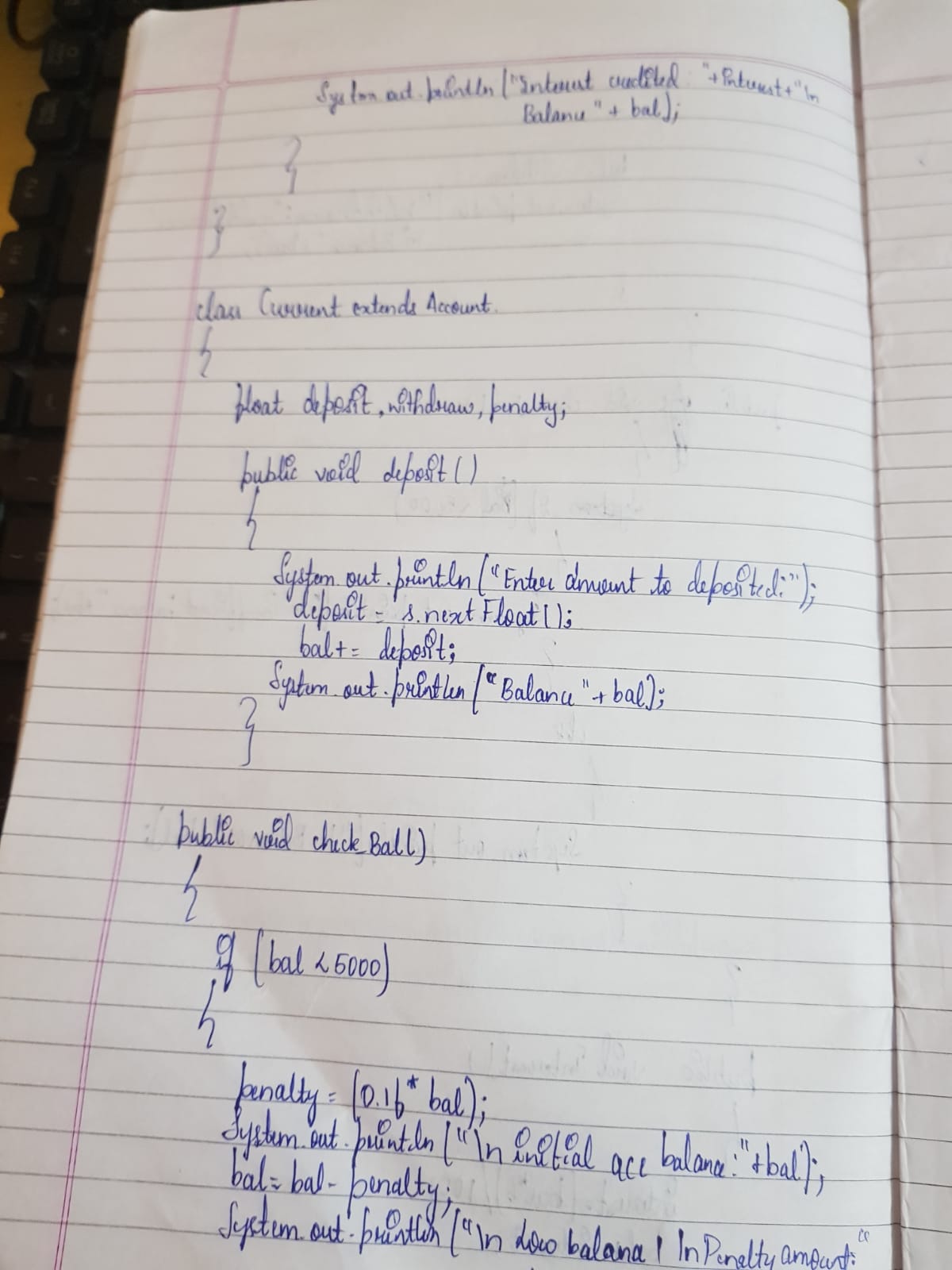
}

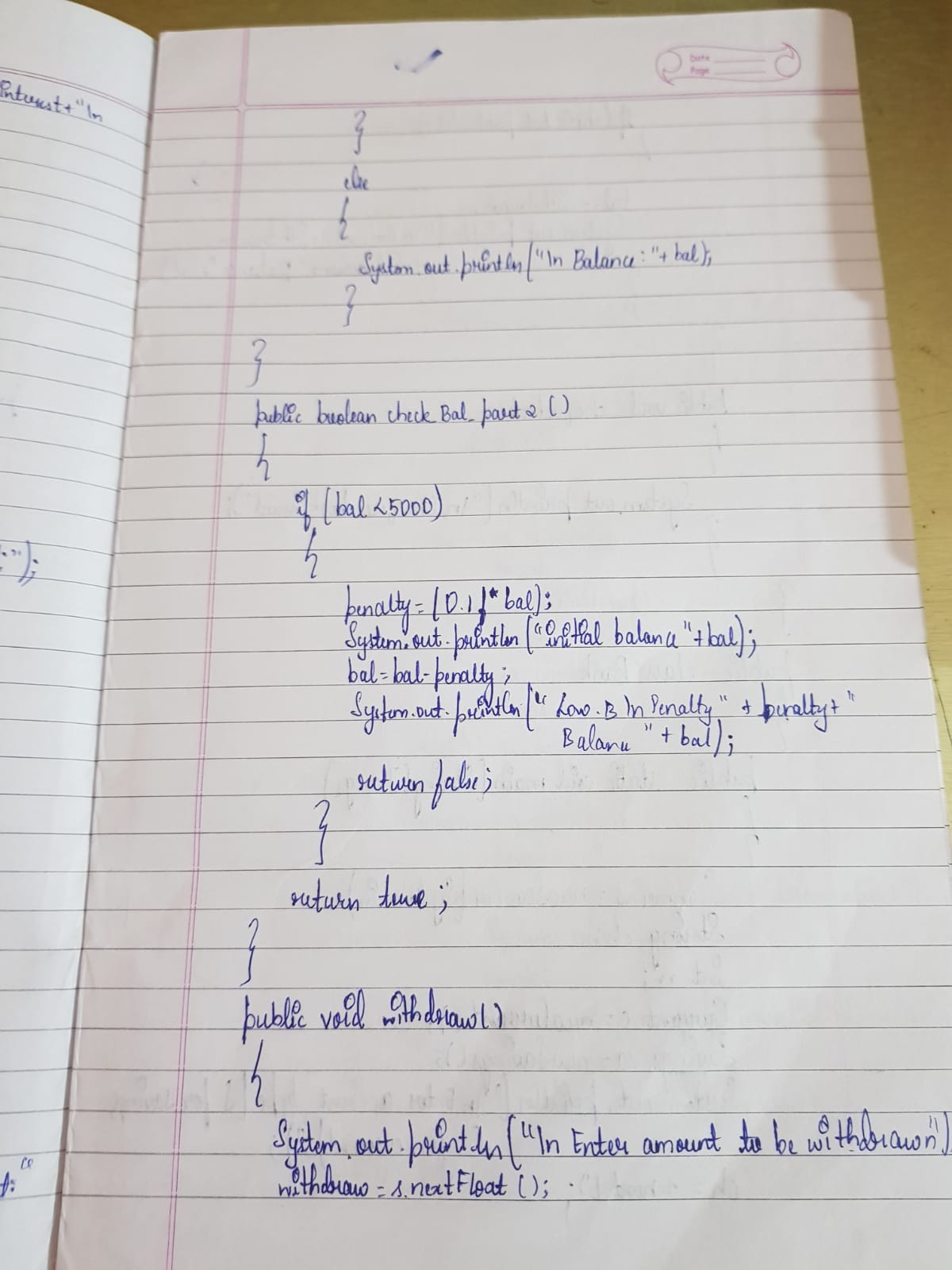
OBSERVATION

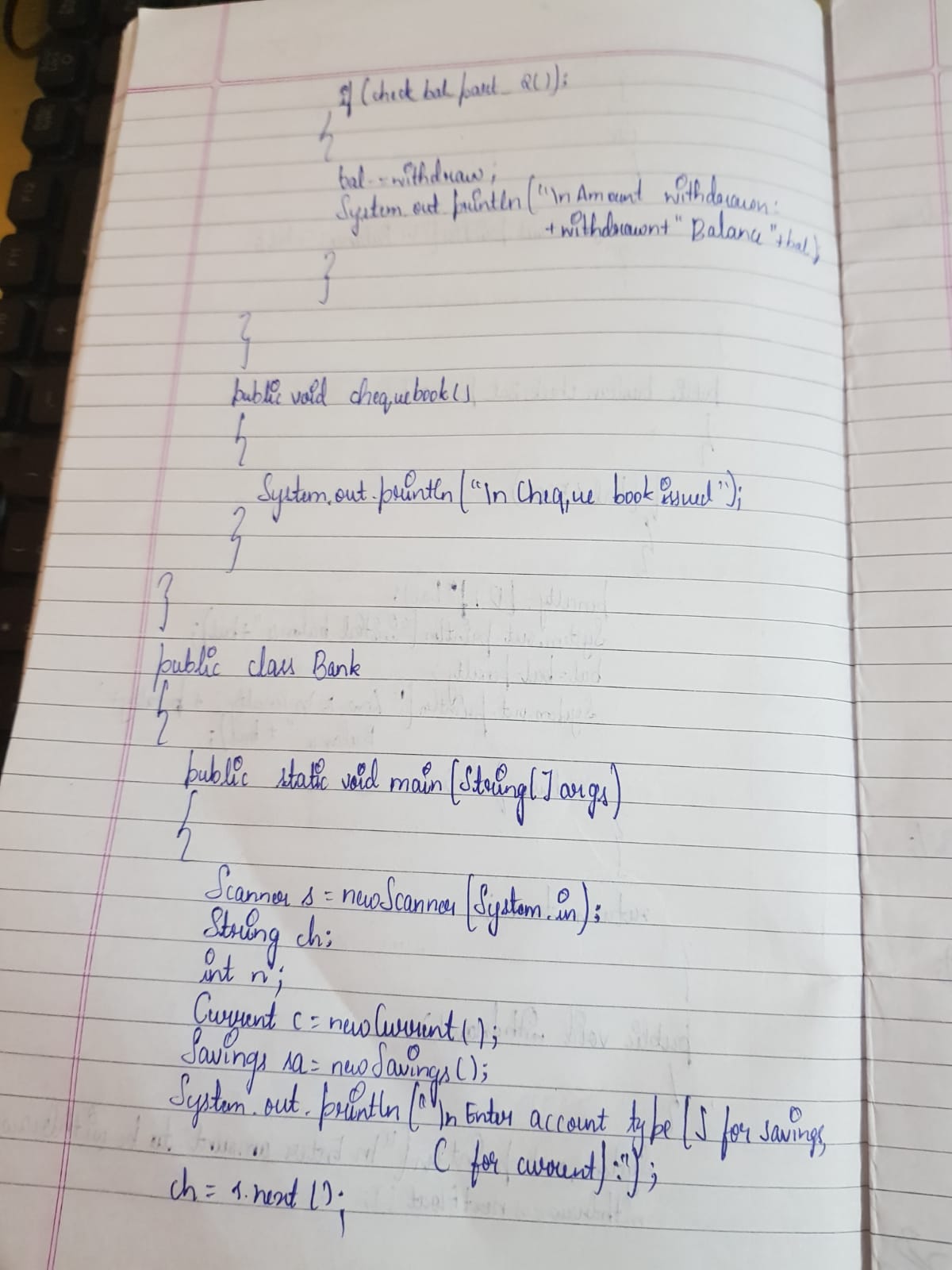


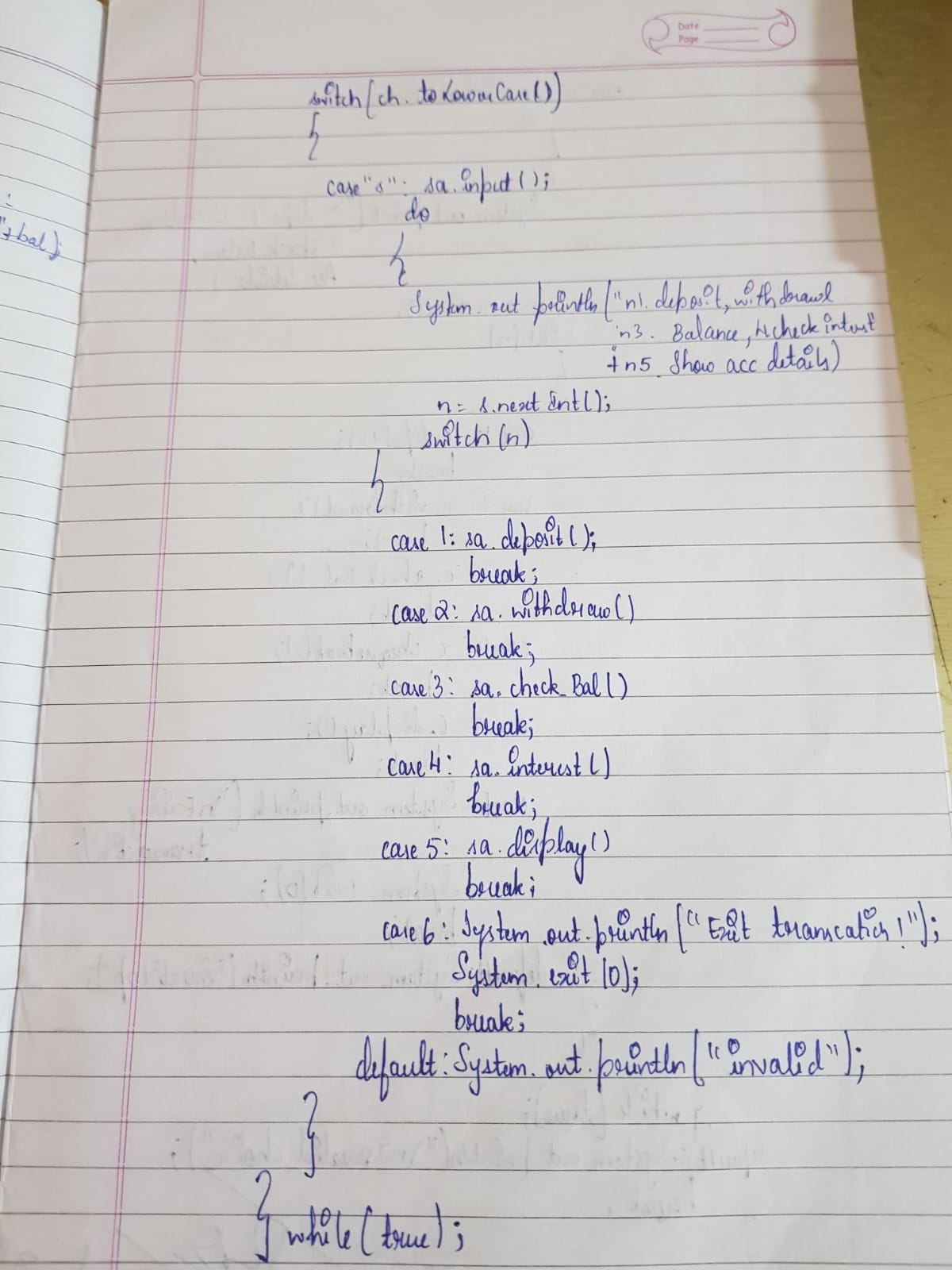


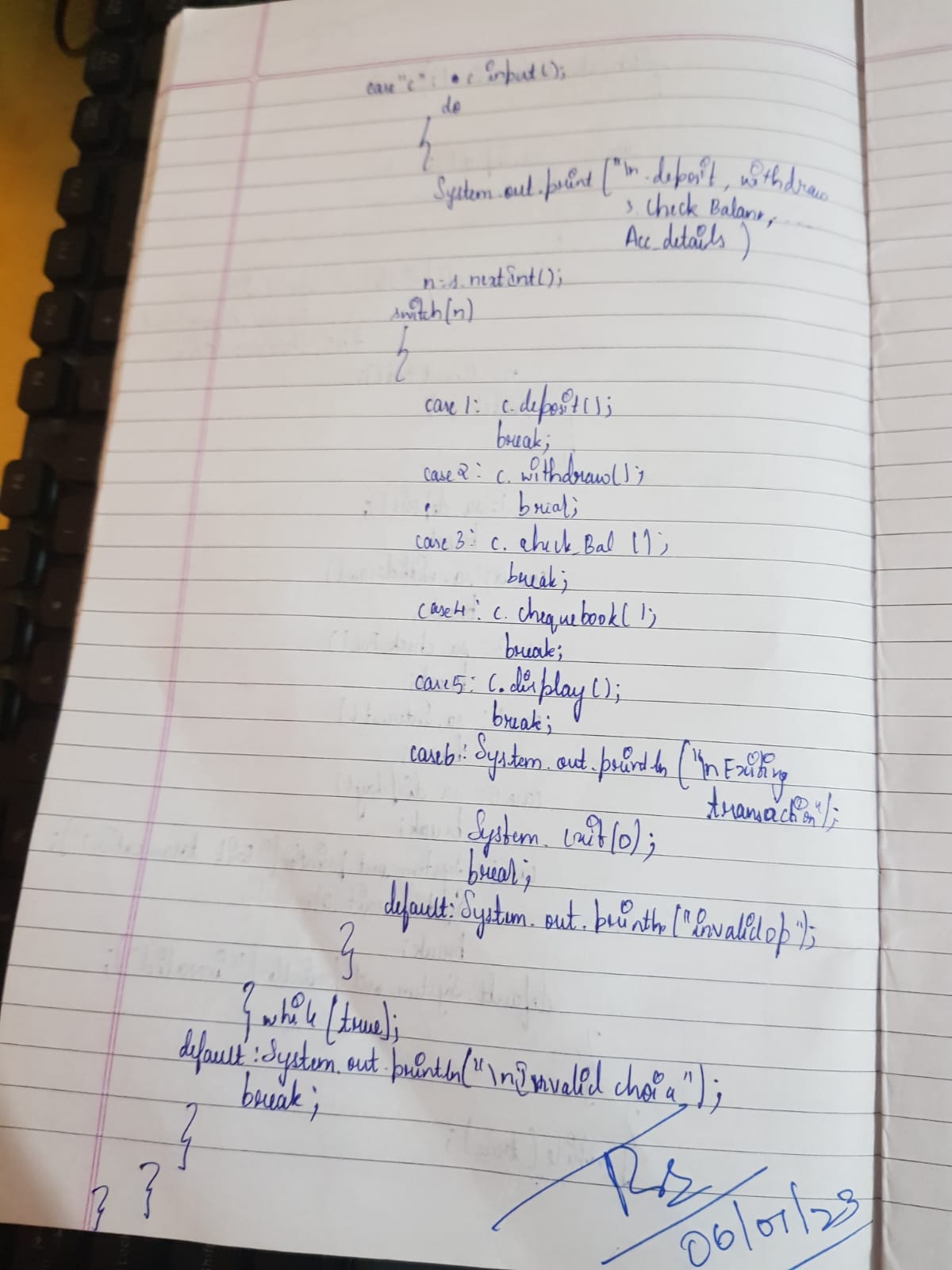




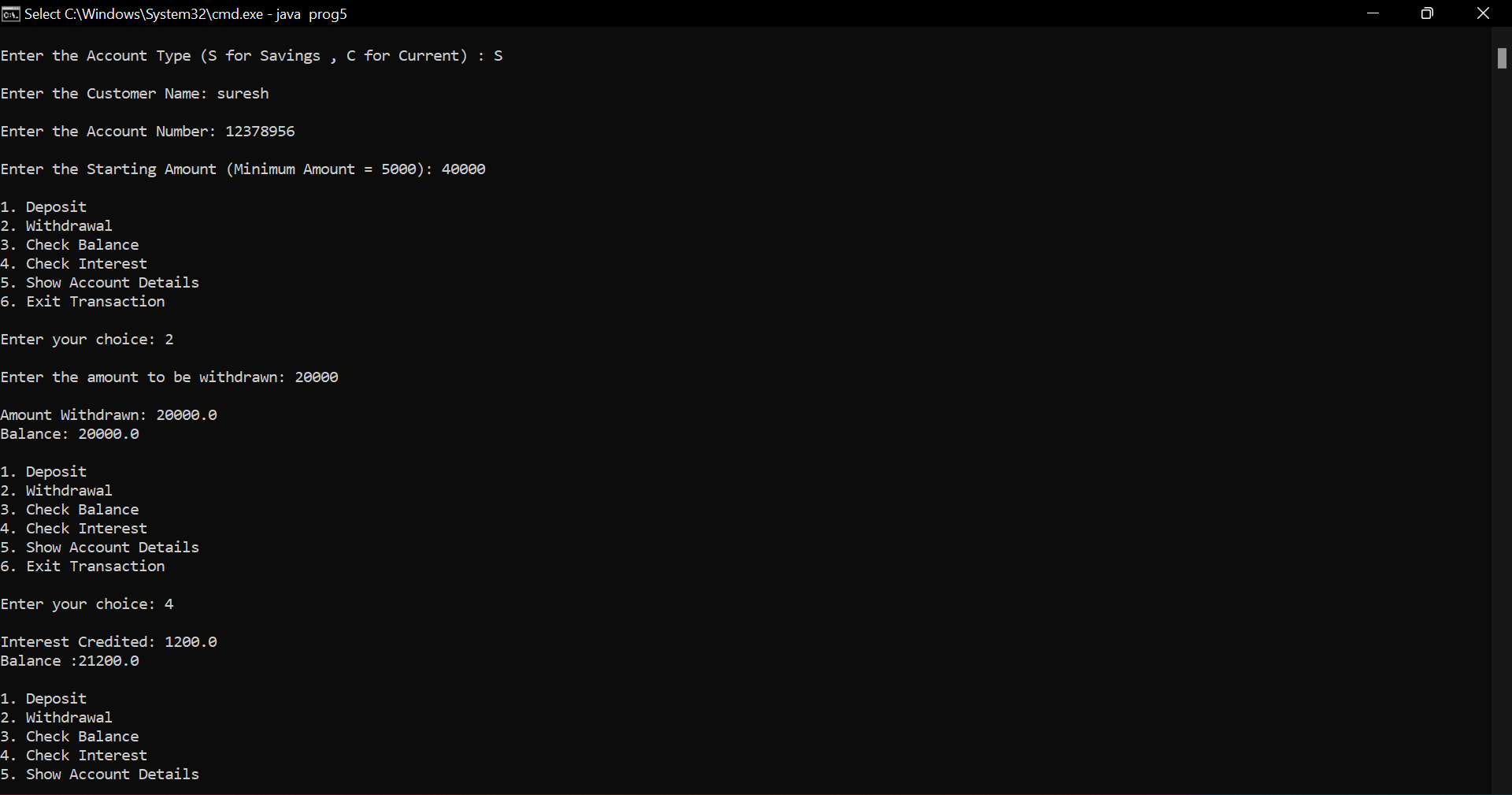


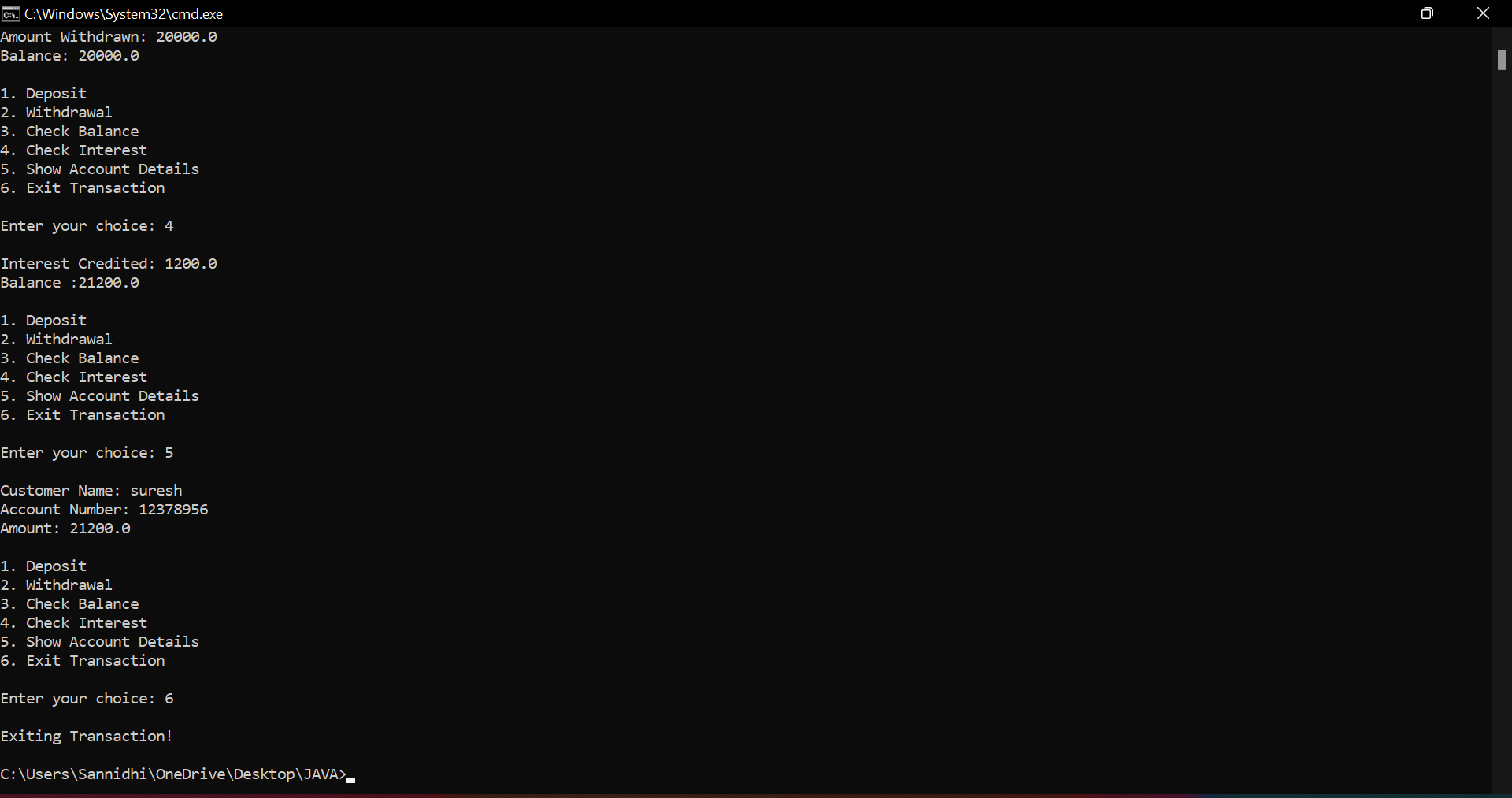






OUTPUT





PROGRAM 6

Anchor College offers both graduate and postgraduate programs.

The college stores the names of the students, their test scores and

the final result for each student. Each student has to take 4 tests in

total. You need to create an application for the college by

implementing the classes based on the class diagram and

description given below.

CODE

import java.util.\*;

interface result

{

void generate\_result();

}

class Student

{

String s\_name;

int t\_score;

String t\_result;

int marr[]=new int[4];

double avg=0;

int flag=1;

Student(){}

Student(String s)

{

s\_name=s;

}

void set\_test\_score(int t\_no,int t\_score) {

marr[t\_no]=t\_score;

}

void set\_studname(String s) {

s\_name=s;

}

void set\_tresult() {

for(int j=0;j<4;j++)

{  avg+=marr[j];

} avg=avg/4; }

double get\_tscore() { set\_tresult(); return avg; }

String get\_tresult() { String p="pass",f="fail"; if(flag==1) { return p; } else return f; }

String get\_studname() { return s\_name; } }

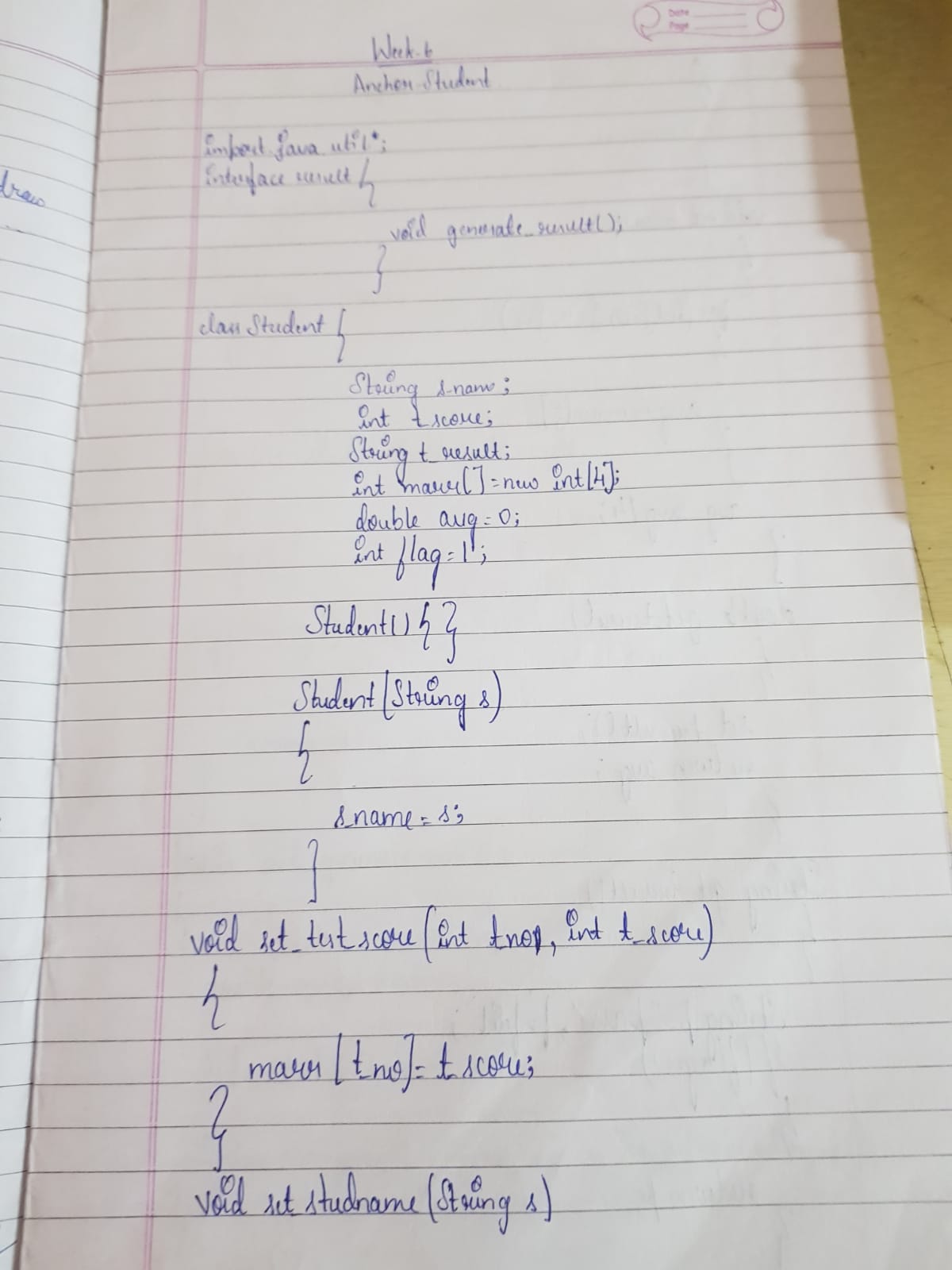
class Ug extends Student implements resultt {

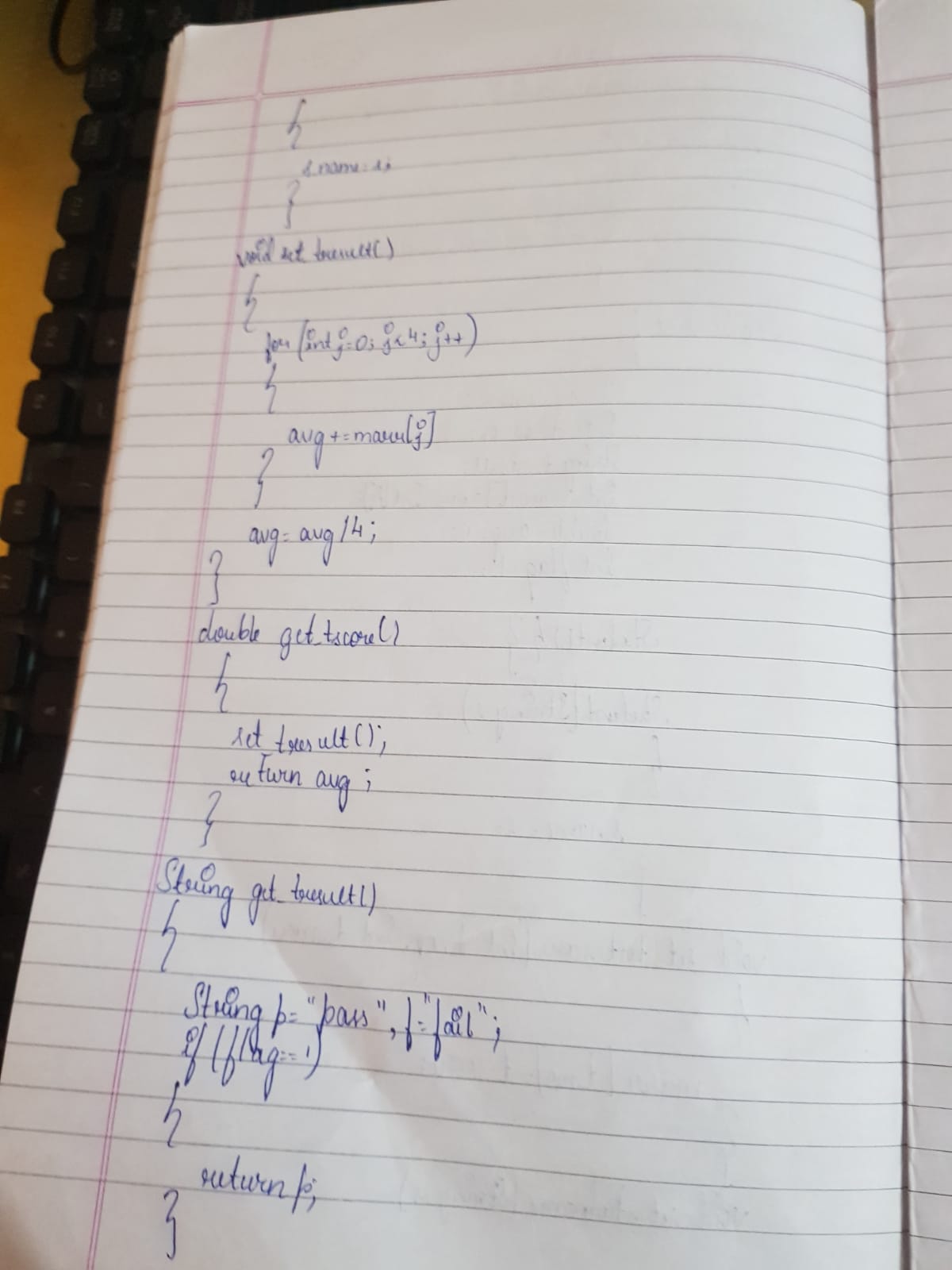
Ug(){} Ug(String s) { s\_name=s; } public void generate\_result() { double q=get\_tscore(); if(q<60) { flag=0;} System.out.println("name"+"     "+get\_studname()); System.out.println("result"+"     "+get\_tresult()); } }

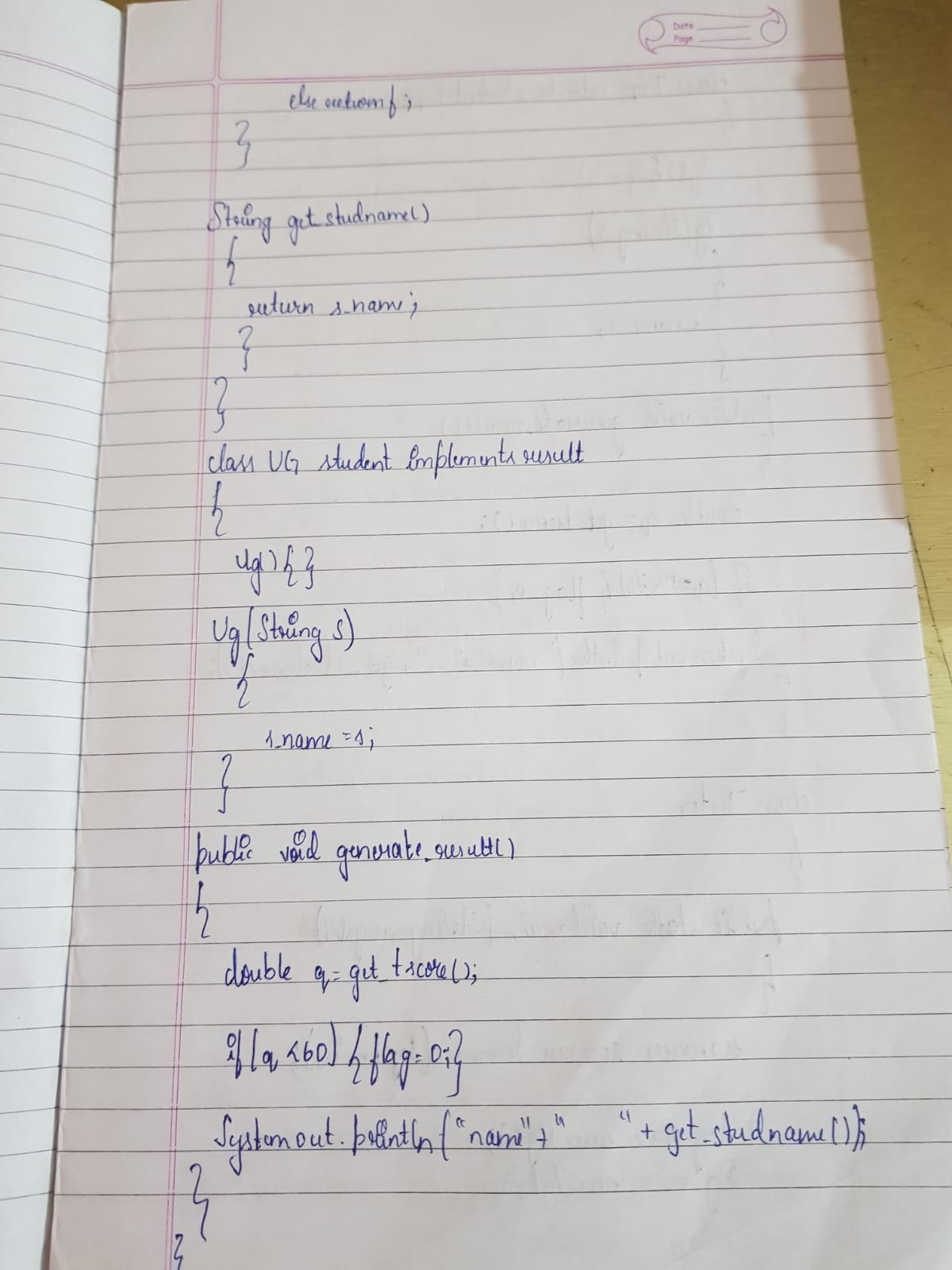
class Pg extends Student implements resultt { Pg(){} Pg(String s) { s\_name=s; } public void generate\_result() { double q=get\_tscore(); if(q<70) { flag=0;} System.out.println("name"+"     "+get\_studname()); System.out.println("result"+"     "+get\_tresult()); } }

class Tester { public static void main(String args[]) { Scanner sc=new Scanner([System.in](http://system.in/)); Ug o2=new Ug(); Pg o3=new Pg(); System.out.println("enter ug student name"); String na=sc.next(); o2.set\_studname(na); System.out.println("enter marks of all 4 tests"); for(int i=0;i<4;i++) { int m=sc.nextInt(); o2.set\_test\_score(i,m); } o2.generate\_result(); System.out.println("enter pg student name"); String nb=sc.next(); o3.set\_studname(nb); System.out.println("enter marks of all 4 tests"); for(int i=0;i<4;i++) { int m=sc.nextInt(); o3.set\_test\_score(i,m); } o3.generate\_result(); } }

OBSERVATION

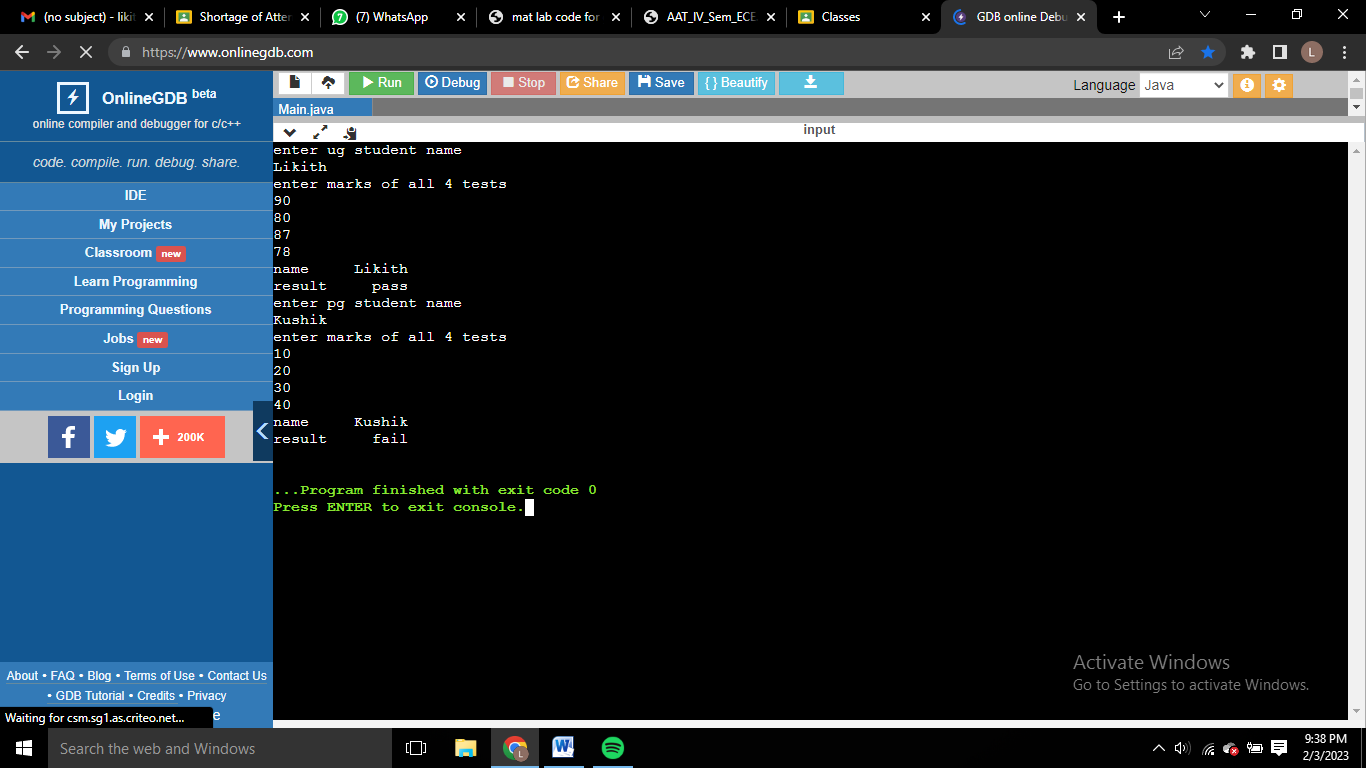








OUTPUT



PROGRAM 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age=father’s age.

CODE

import java.util.Scanner;

class WrongAge extends Exception

{

public String getMessage()

{

return "Age Cannot Be Negative";

}

}

class InvalidAge extends Exception

{

public String getMessage()

{

return "Son's Age cannot be greater than Father's!";

}

}

class Father

{

Scanner s = new Scanner(System.in);

int f;

Father() throws WrongAge

{

System.out.print("Enter the Father's Age: ");

f = s.nextInt();

try

{

if(f<0)

throw new WrongAge();

}

catch(WrongAge e1)

{

System.out.println(e1.getMessage());

System.exit(0);

}

}

}

class Son extends Father

{

int son;

Son() throws WrongAge,InvalidAge

{

super();

System.out.print("Enter the Son's Age: ");

son = s.nextInt();

try

{

if(son<0)

throw new WrongAge();

}

catch(WrongAge e2)

{

System.out.println(e2.getMessage());

}

try

{

if(son>f)

throw new InvalidAge();

}

catch(InvalidAge e3)

{

System.out.println(e3.getMessage());

}

}

}

public class prog7

{

public static void main(String[] args) throws WrongAge,InvalidAge

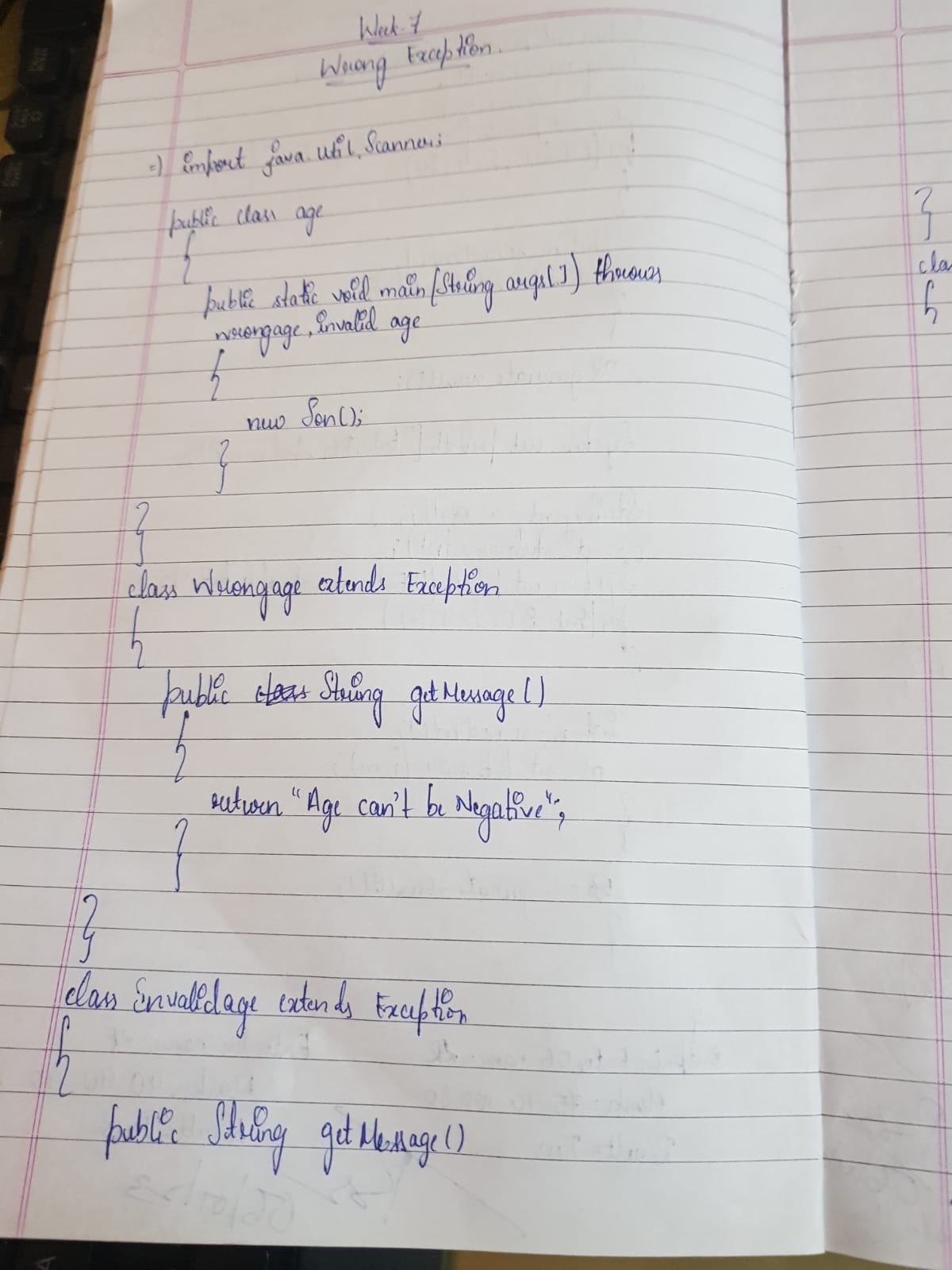
{

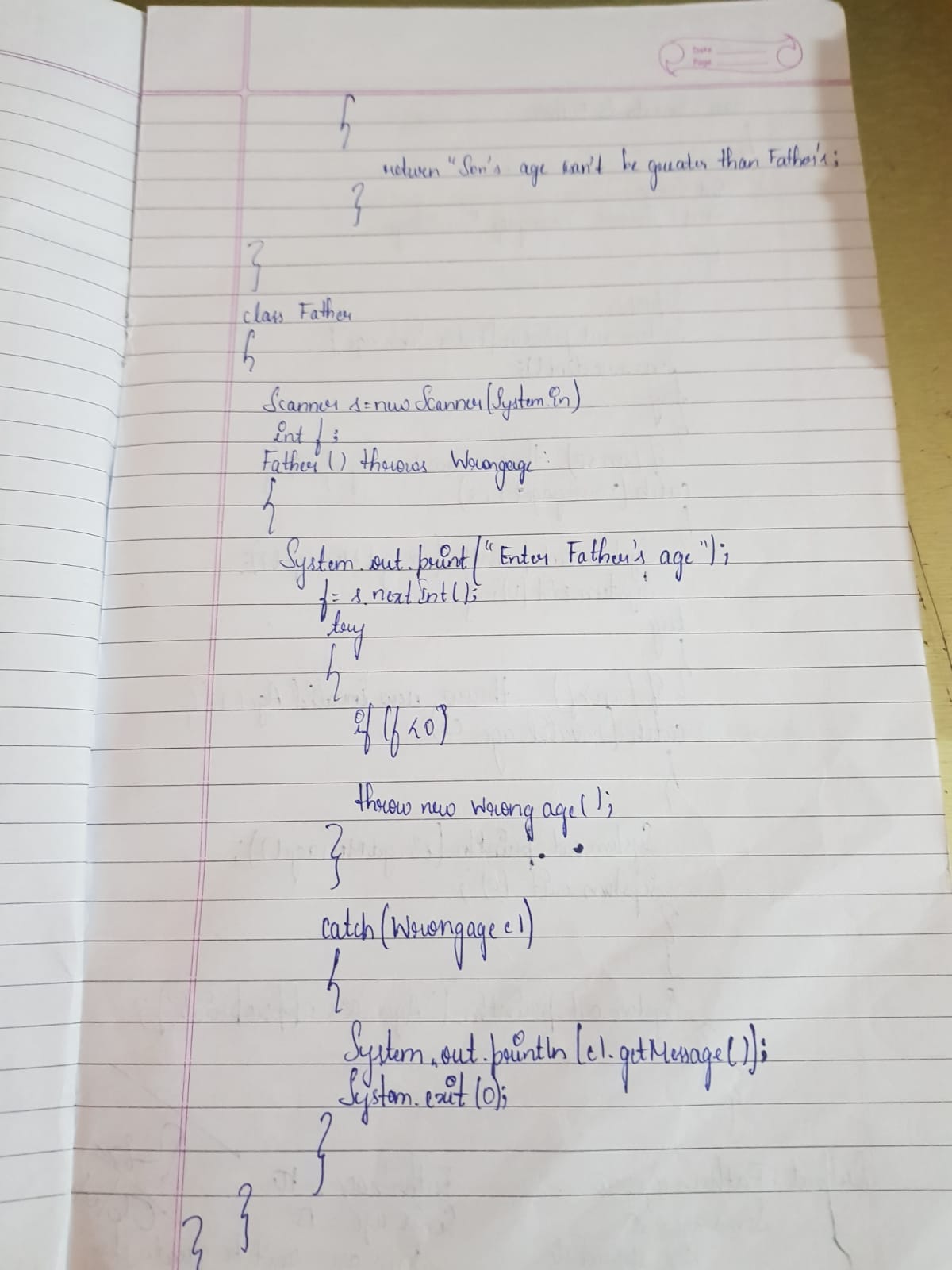
new Son();

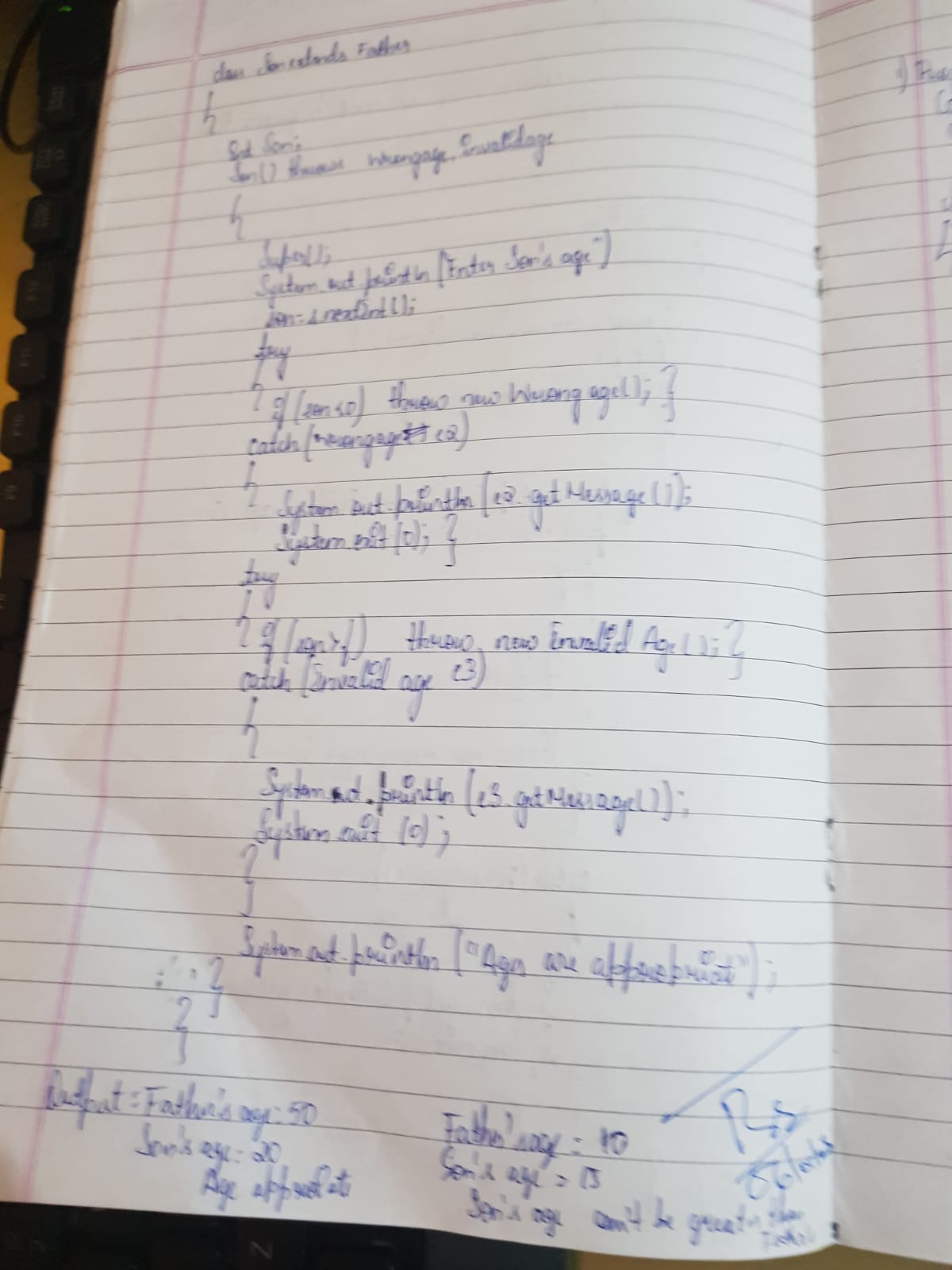
}

}

OBSERVATION







OUTPUT



PROGRAM 8

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

CODE

class Thread\_1 extends Thread

{

public void run()

{

int i = 0;

while(i<100)

{

try

{

Thread.sleep(10000);

System.out.println("BMSCE");

}

catch(Exception e)

{

System.out.println("Exception: "+e);

}

i++;

}

}

}

class Thread\_2 extends Thread

{

public void run()

{

int i = 0;

while(i<100)

{

try

{

Thread.sleep(2000);

System.out.println("CSE");

}

catch(Exception e)

{

System.out.println("Exception "+e);

}

i++;

}

}

}

public class prog8

{

public static void main(String[] args)

{

Thread t1 = new Thread\_1();

Thread t2 = new Thread\_2();

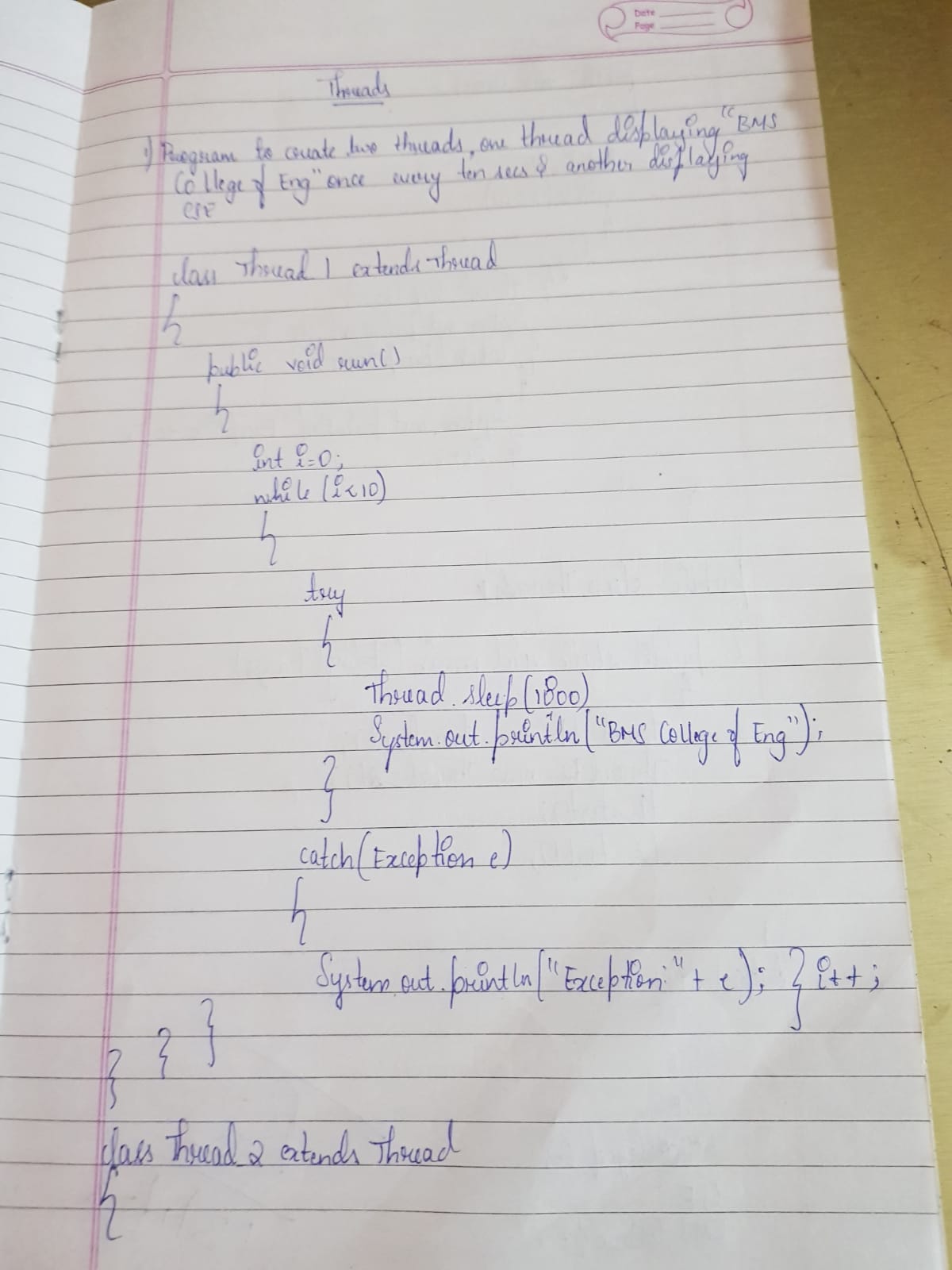
t1.start();

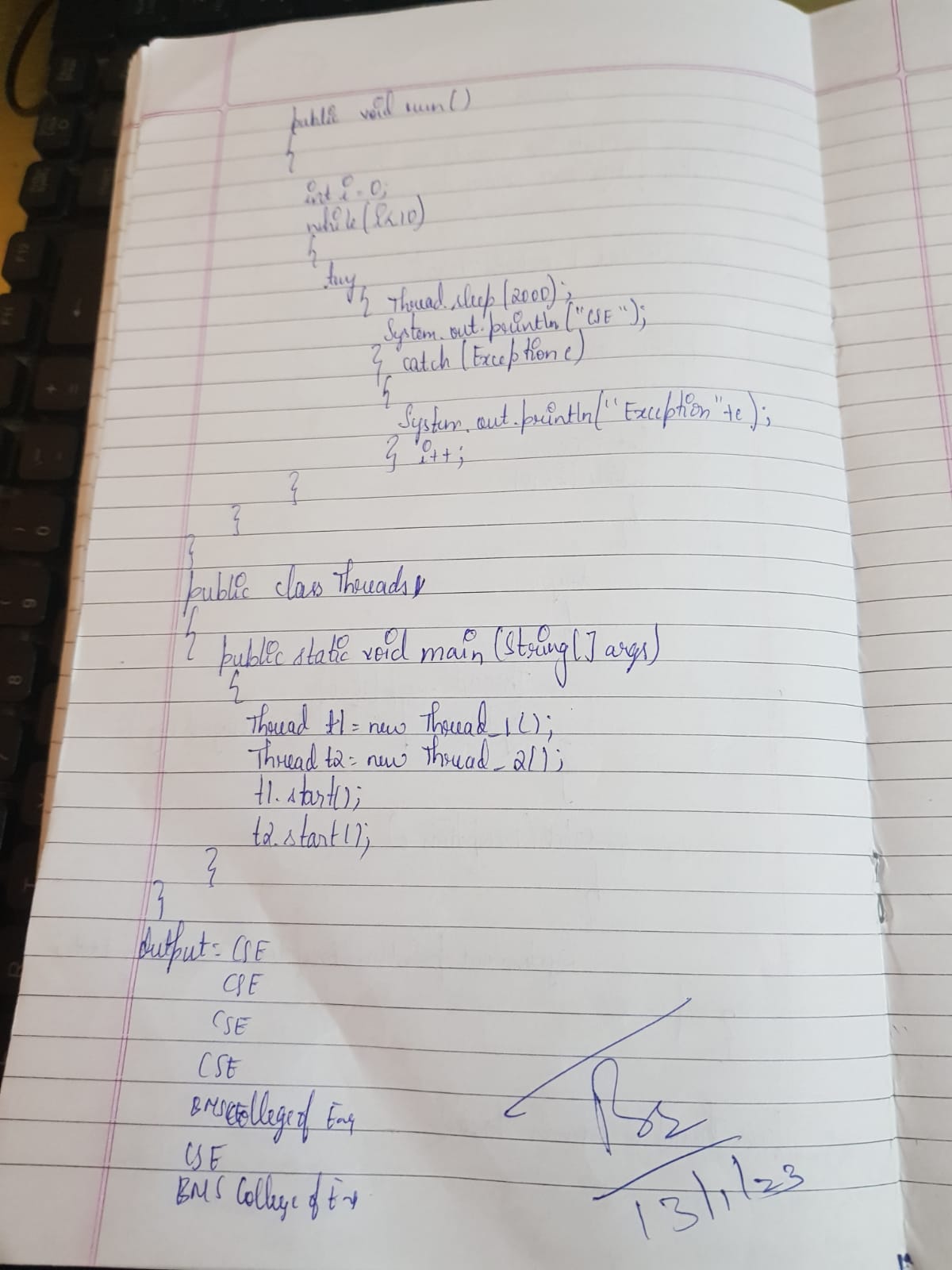
t2.start();

}

}

OBSERVATION





OUTPUT

