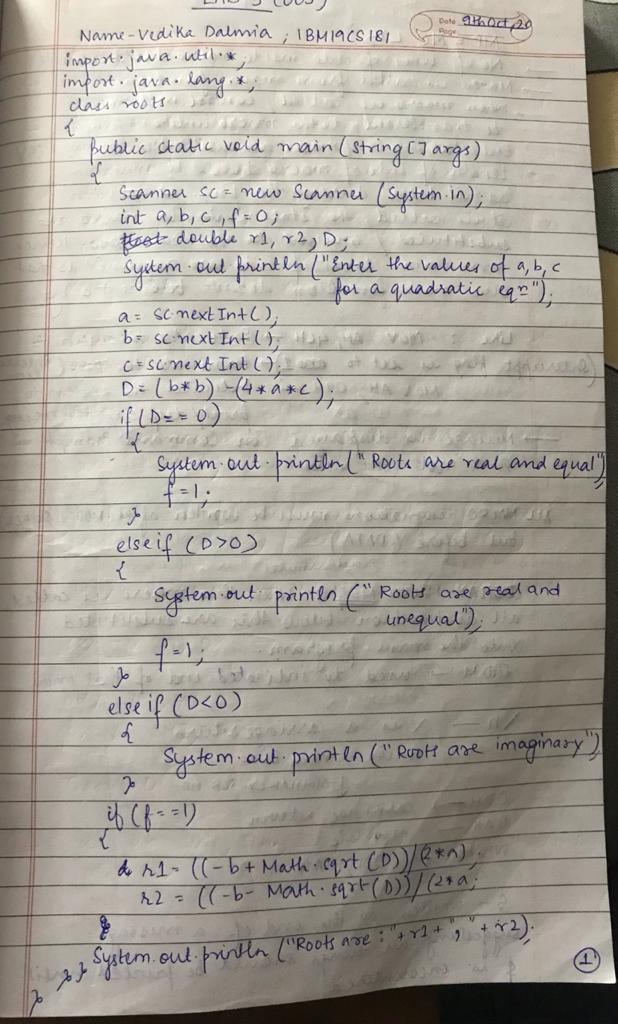
**NAME-VEDIKA DALMIA; USN-1BM19CS181;SEC-3D;LAB BATCH-1**

**LAB PROGRAM-1:**

Q.) 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.

**WRITEUP:**

**PROGRAM**

import java.util.\*;

class Roots

{

public static void main(String[] args)

{

int a,b,c,f=0;

double D;

Scanner sc=new Scanner(System.in);

System.out.println("\nEnter the values of a,b,c:");

a=sc.nextInt();

b=sc.nextInt();

c=sc.nextInt();

D=(b\*b)-(4\*a\*c);

if(D==0)

{

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

f=1;

}

else if(D>0)

{

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

f=1;

}

else if(D<0)

{

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

}

if(f==1)

{

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

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

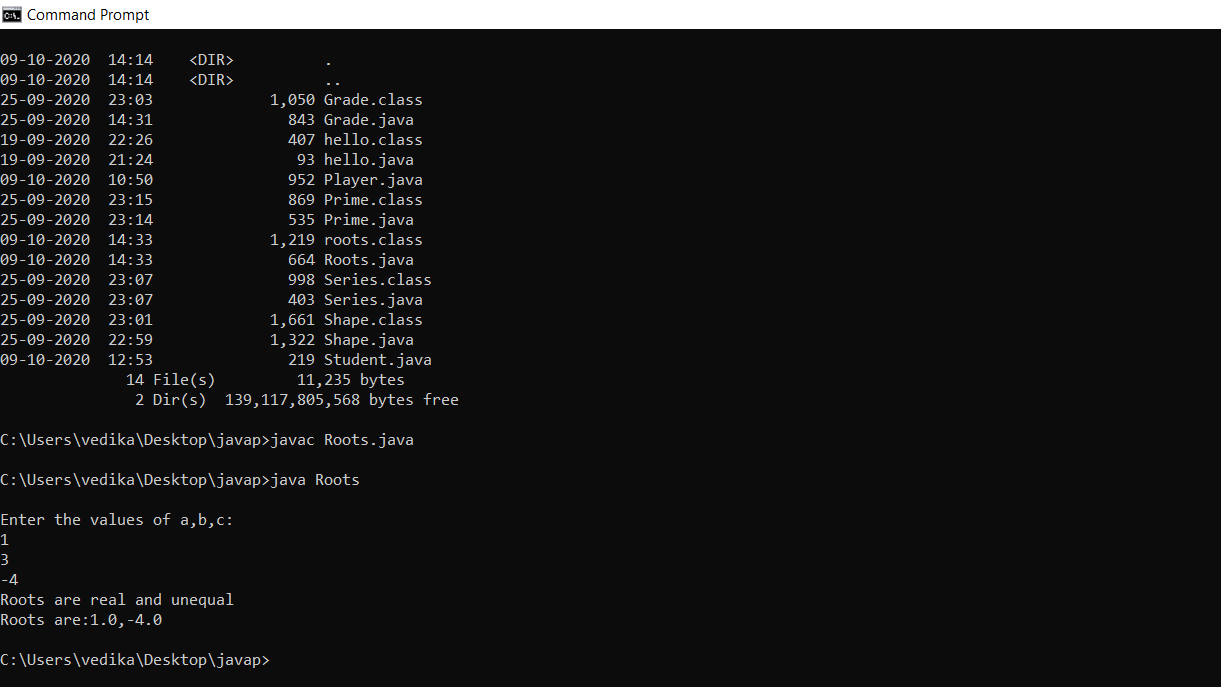
System.out.println("Roots are:"+r1+","+r2);

}

}

}

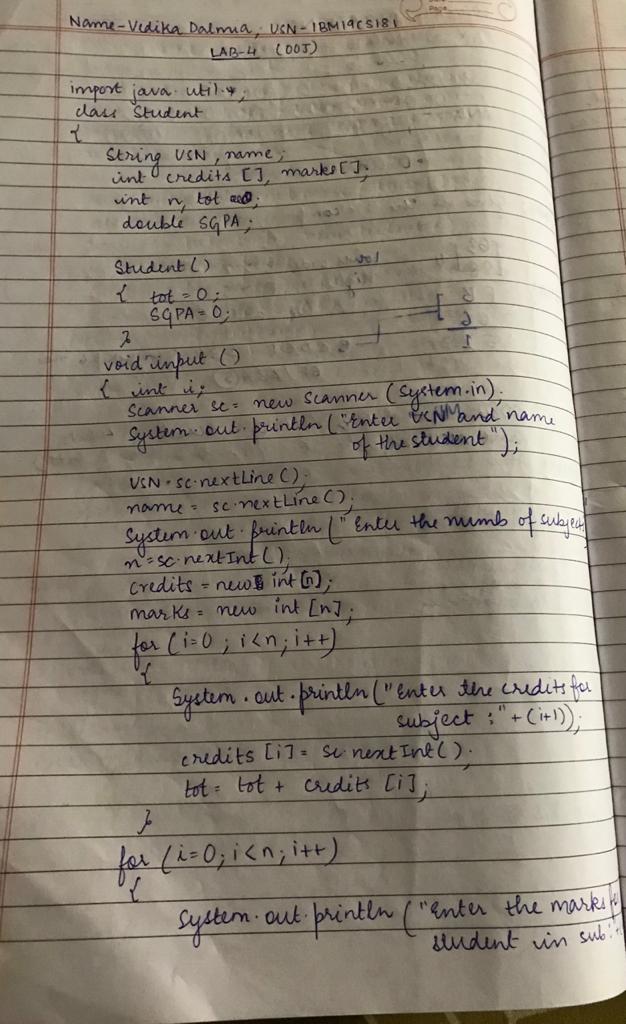
**OUTPUT SCREEN:1**

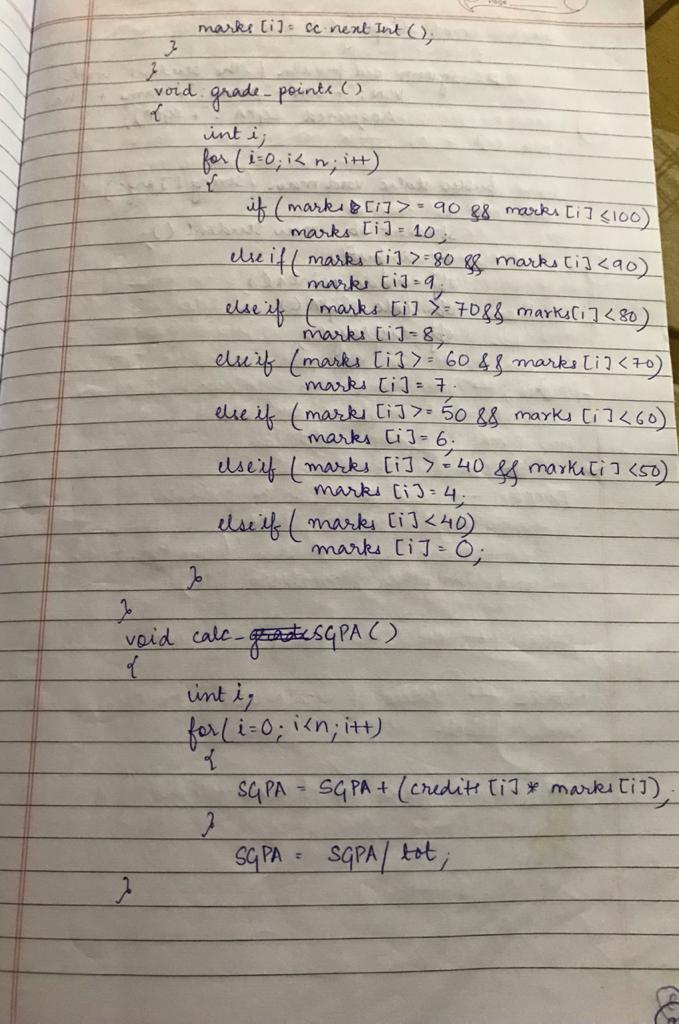


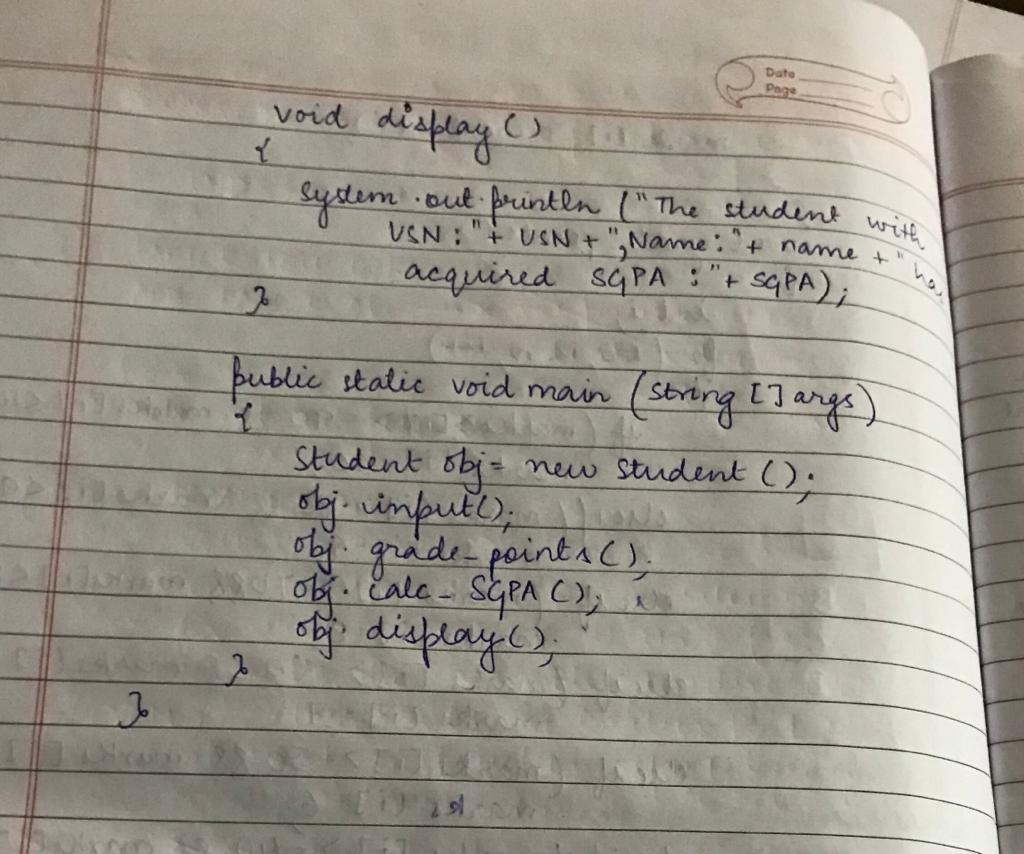
**LAB PROGRAM 2:**

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.

**WRITEUP:**







**PROGRAM**

import java.util.\*;

class Student2

{

String USN;

String name;

int credits[];

int marks[];

int n,tot;

double SGPA;

Student2()

{

tot=0;

SGPA=0;

}

void input()

{

Scanner sc=new Scanner(System.in);

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

USN=sc.nextLine();

name=sc.nextLine();

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

n=sc.nextInt();

credits=new int[n];

marks =new int[n];

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

{

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

credits[i]=sc.nextInt();

tot=tot+credits[i];

}

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

{

System.out.println("Enter the marks of the student for subject:"+(i+1));

marks[i]=sc.nextInt();

}

}

void grade\_points()

{

int i;

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

{

if(marks[i]>=90 && marks[i]<100)

{

marks[i]=10;

}

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

{

marks[i]=9;

}

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

{

marks[i]=8;

}

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

{

marks[i]=7;

}

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

{

marks[i]=6;

}

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

{

marks[i]=4;

}

else if(marks[i]<40)

{

marks[i]=0;

}

}

}

void calc\_SGPA()

{

int i;

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

{

SGPA=SGPA+(credits[i]\*marks[i]);

}

SGPA=SGPA/tot;

}

void display()

{

System.out.println("The student with USN:"+USN+", Name:"+name+" has SGPA:"+SGPA);

}

public static void main(String[] args)

{

Student2 obj=new Student2();

obj.input();

obj.grade\_points();

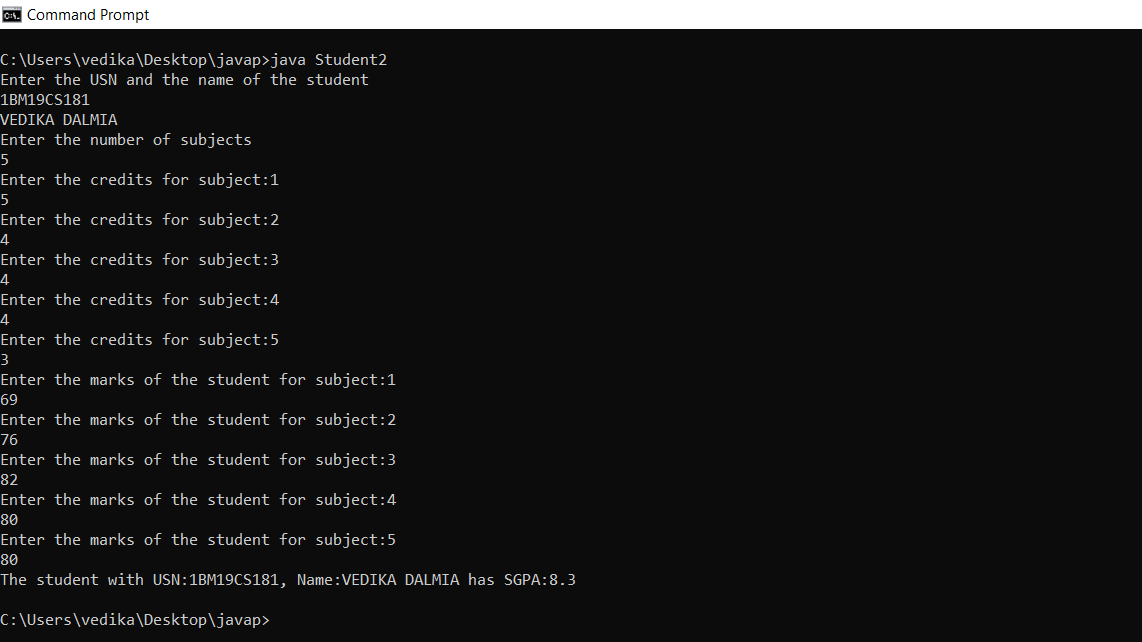
obj.calc\_SGPA();

obj.display();

}

}

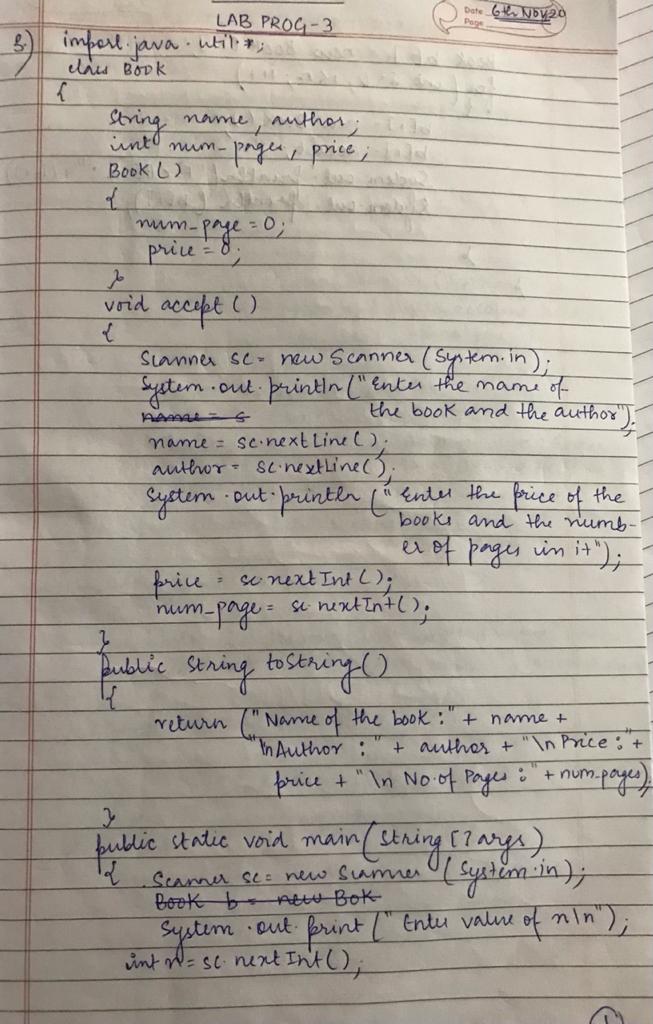
**OUTPUT SCREEN:2**

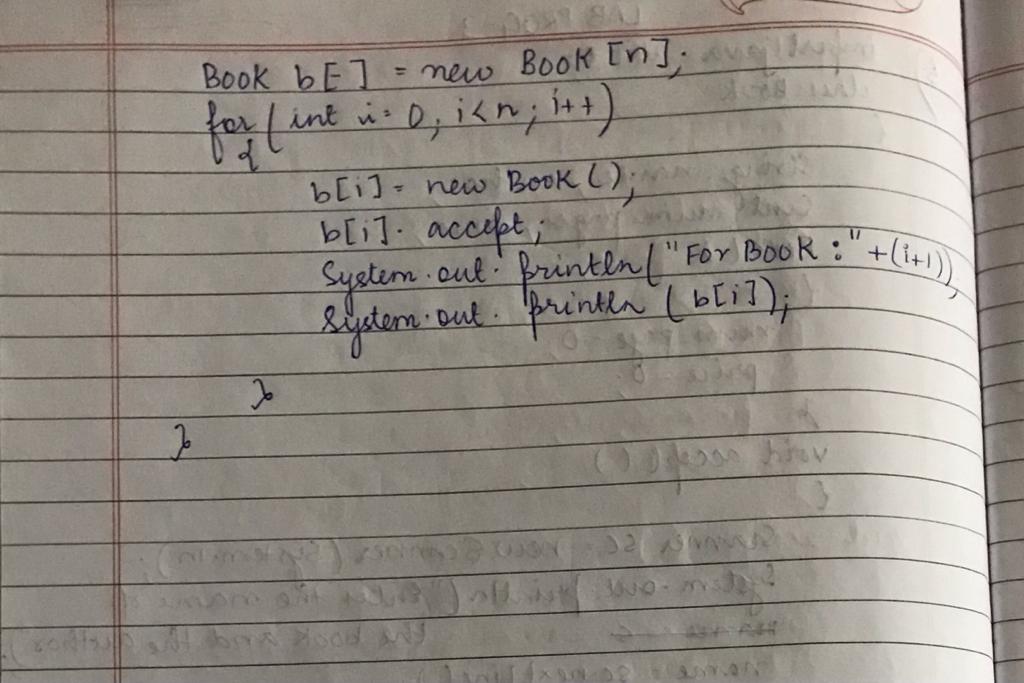


**LAB PROGRAM 3:**

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.

**WRITEUP:**





**PROGRAM**

import java.util.Scanner;

class Book

{

private String name,author;

private double price;

private int num\_pages;

Book()

{

name="A";

author="BCD";

price=340.0;

num\_pages=500;

}

void Input()

{

Scanner sc=new Scanner(System.in);

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

name=sc.nextLine();

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

author=sc.nextLine();

System.out.println("Enter the the no.of pages: ");

num\_pages=sc.nextInt();

System.out.println("Enter the price: ");

price=sc.nextDouble();

}

public String toString()

{

String temp="Book name: "+name+"\nAuthor name: "+author+"\nNo.of pages: "+num\_pages+"\nPrice: "+price+"\n";

return(temp);

}

}

class B

{

public static void main(String args[])

{

int i,n;

Scanner sc=new Scanner(System.in);

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

n=sc.nextInt();

Book[] obj=new Book[n];

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

{

obj[i]=new Book();

}

System.out.println("\t\t\*\*Enter Book Details\*\*");

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

{

System.out.println("\nBook "+(i+1)+";");

obj[i].Input();

}

System.out.println("Book Details:");

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

{

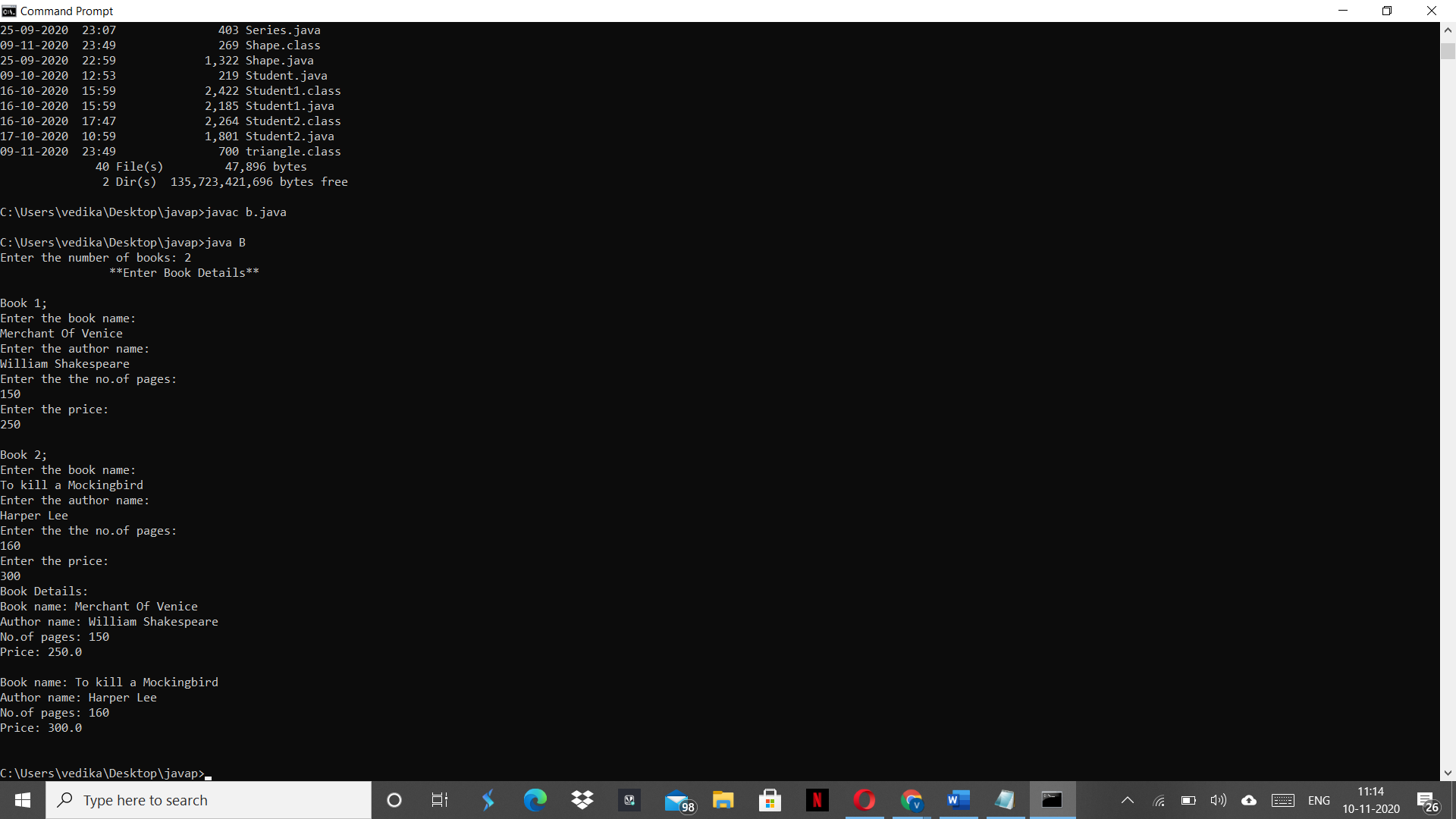
System.out.println(obj[i]);

}

}

}

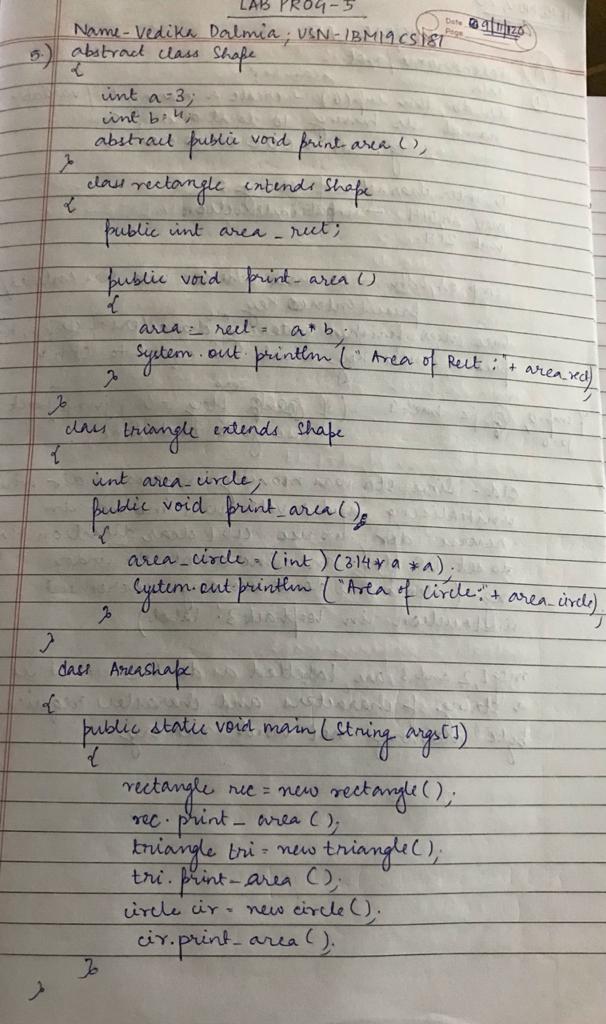
**OUTPUT SCREEEN:3**



**LAB 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.

**WRITEUP:**



**PROGRAM**

abstract class Shape

{

int a=3;

int b=4;

abstract public void print\_area();

}

class rectangle extends Shape

{

public int area\_rect;

public void print\_area()

{

area\_rect=a\*b;

System.out.println("The area of rectangle is: "+area\_rect);

}

}

class triangle extends Shape

{

int area\_tri;

public void print\_area()

{

area\_tri=(int) (0.5\*a\*b);

System.out.println("The area of triangle is: "+area\_tri);

}

}

class circle extends Shape

{

int area\_circle;

public void print\_area()

{

area\_circle=(int) (3.14\*a\*a);

System.out.println("The area of circle is: "+area\_circle);

}

}

class abs{

public static void main(String[] args){

rectangle rec = new rectangle();

rec.print\_area();

triangle tri = new triangle();

tri.print\_area();

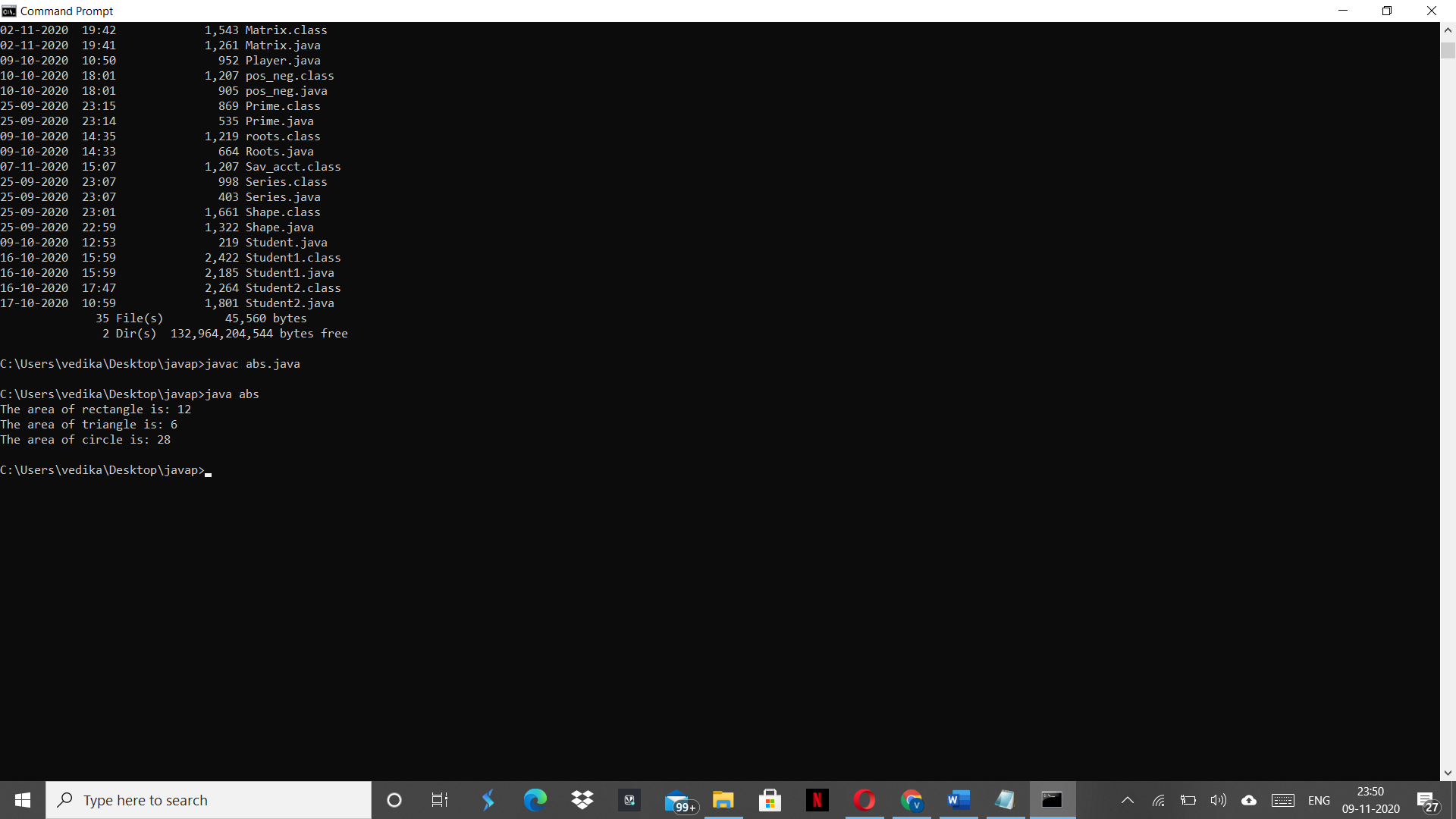
circle cir = new circle();

cir.print\_area();

}

}

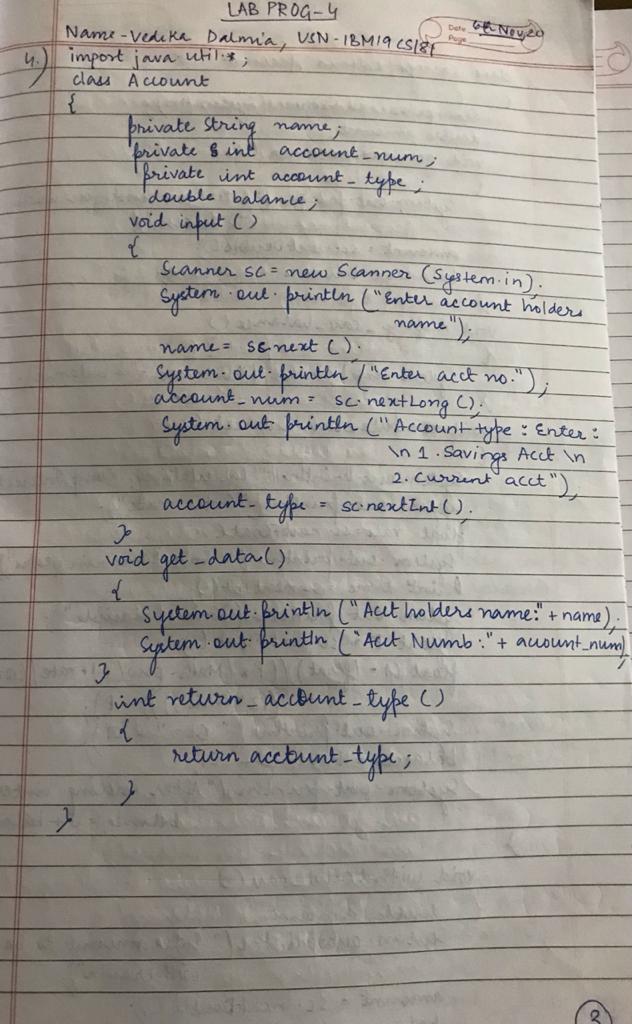
**OUTPUT SCREEN:4**

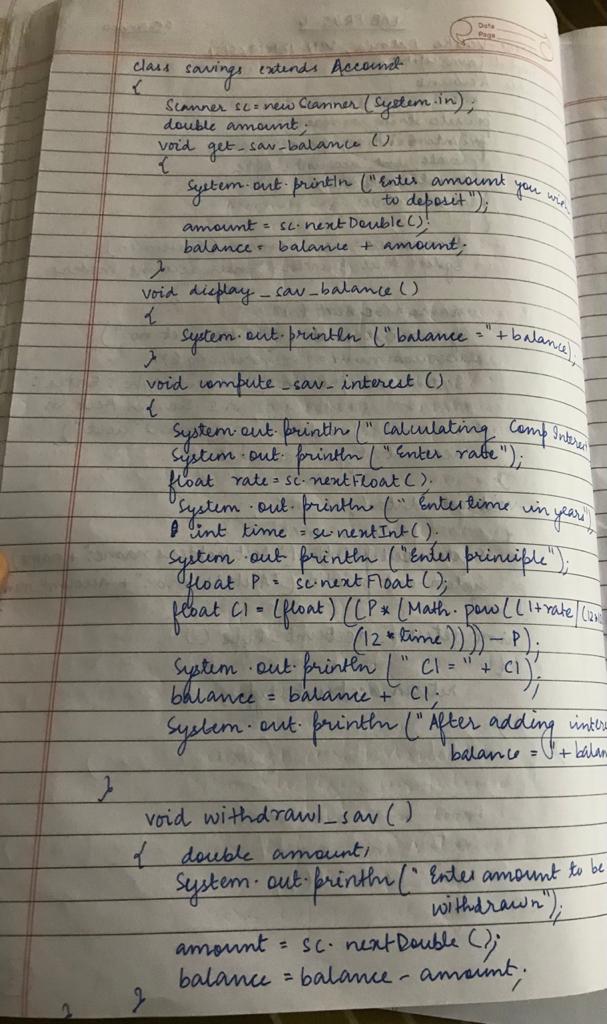


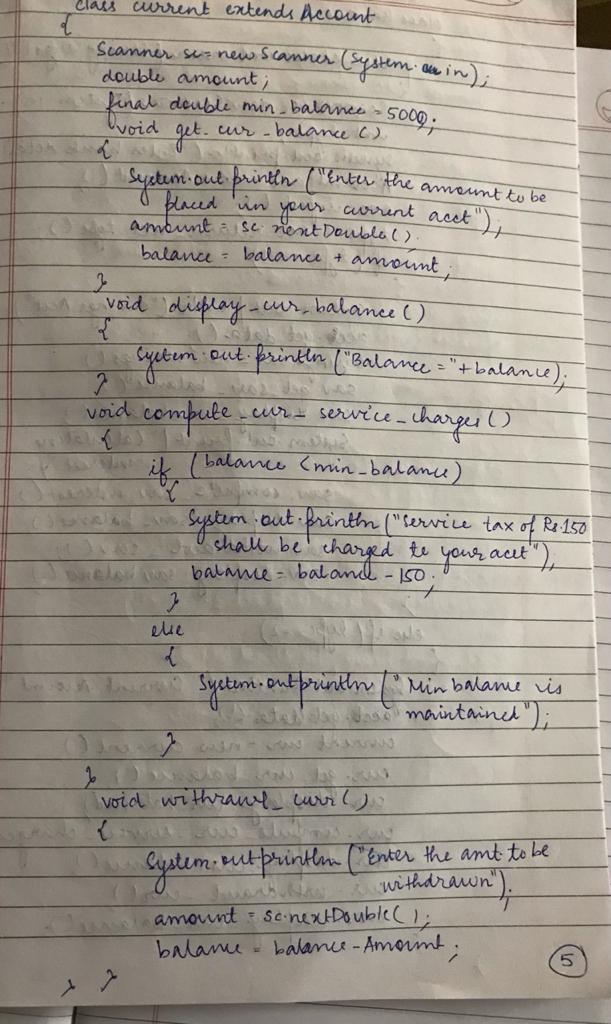
**LAB PROGRAM 5:**

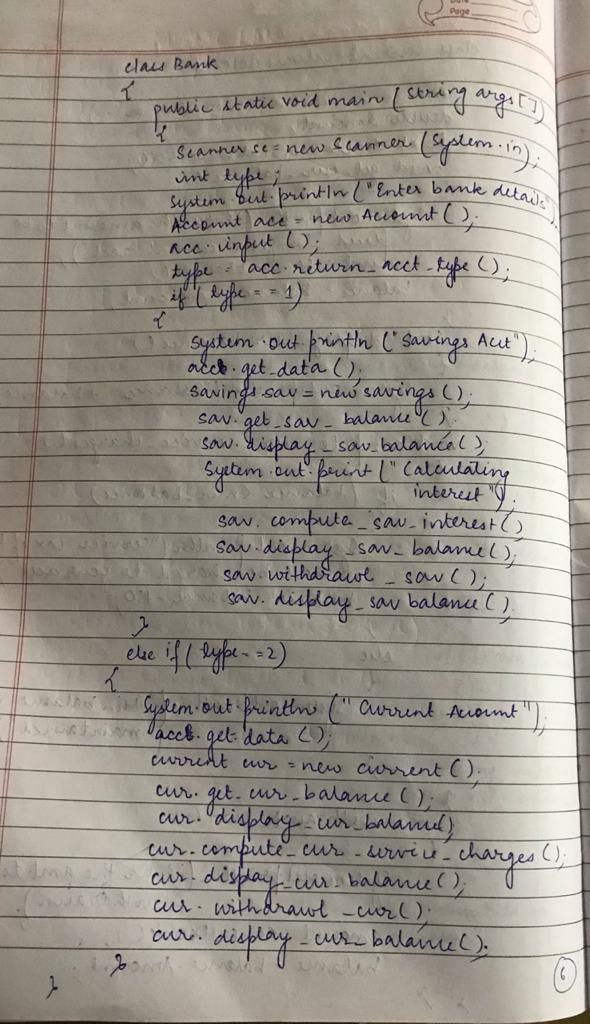
Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book 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 Curr-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.

**WRITEUP:**









**PROGRAM**

import java.util.Scanner;

class Account

{

private String name;

private long account\_number;

private int account\_type;

double balance;

void Input()

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter Account Holder Name");

name = sc.nextLine();

System.out.println("Enter the account Number");

account\_number=sc.nextLong();

System.out.println("Choose the account type:\n1.savings account\n2.current account");

account\_type=sc.nextInt();

}

void get\_data(){

System.out.println("Account Holder: "+name);

System.out.println("Account Number: "+account\_number);

}

int return\_account\_type()

{

return account\_type;

}

}

class savings extends Account

{

Scanner sc=new Scanner(System.in);

double amount;

void get\_sav\_balance()

{

System.out.println("Enter the Amount to be placed in your Savings Account");

amount=sc.nextDouble();

balance+=amount;

}

void display\_sav\_blnce()

{

System.out.println("balance= "+balance);

}

void compute\_sav\_interest()

{

System.out.println("\n\*\*\*Calculating Compound Interest\*\*\*");

System.out.println("Enter annual interest rate: ");

float rate = sc.nextFloat();

System.out.println("Enter time in years: ");

float time = sc.nextFloat();

System.out.println("Enter principle: ");

float principle = sc.nextFloat();

float CI = (float)((principle\*(Math.pow((1 + rate / (12\*100)),(12\*time))))-principle);

System.out.println("The Compound Interest is: " +CI);

balance = balance+CI;

System.out.println("Balance after adding Interest: "+balance);

}

void withdrawl\_sav()

{

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

amount = sc.nextDouble();

balance=balance-amount;

}

}

class current extends Account

{

Scanner sc = new Scanner(System.in);

double amount;

final double min\_balance=500;

void get\_cur\_balance()

{

System.out.println("Enter the amount to be placed in your current account");

amount=sc.nextDouble();

balance+=amount;

}

void display\_cur\_blnce()

{

System.out.println("Balance = "+balance);

}

void compute\_cur\_service\_charges()

{

if(balance<min\_balance)

{

System.out.println("service tax of rs.100 shall be levied");

balance=balance-100;

}

else

{

System.out.println("Minimum balance is Maintained");

}

}

void withdrawl\_cur()

{

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

amount=sc.nextDouble();

balance=balance-amount;

}

}

class BankF

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int type;

System.out.println("Enter the bank details");

Account acc=new Account();

acc.Input();

type=acc.return\_account\_type();

if (type==1)

{

System.out.println("SAVINGS ACCOUNT");

acc.get\_data();

savings sav = new savings();

sav.get\_sav\_balance();

sav.display\_sav\_blnce();

System.out.println("Calculating Interest:\n");

sav.compute\_sav\_interest();

sav.display\_sav\_blnce();

sav.withdrawl\_sav();

sav.display\_sav\_blnce();

}

if(type==2)

{

System.out.println("CURRENT ACCOUNT");

acc.get\_data();

current cur=new current();

cur.get\_cur\_balance();

cur.display\_cur\_blnce();

cur.compute\_cur\_service\_charges();

cur.display\_cur\_blnce();

cur.withdrawl\_cur();

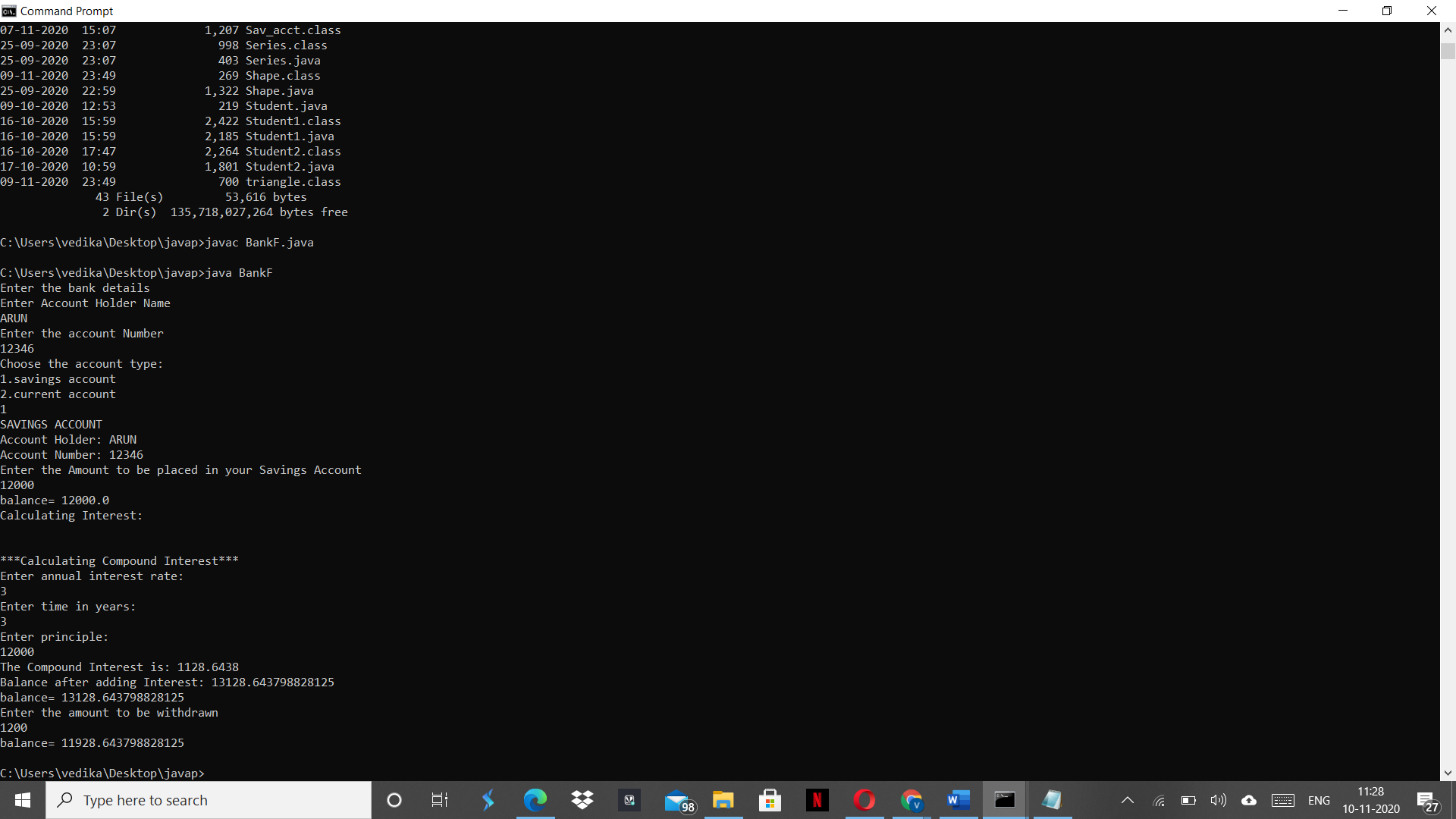
cur.display\_cur\_blnce();

}

}

}

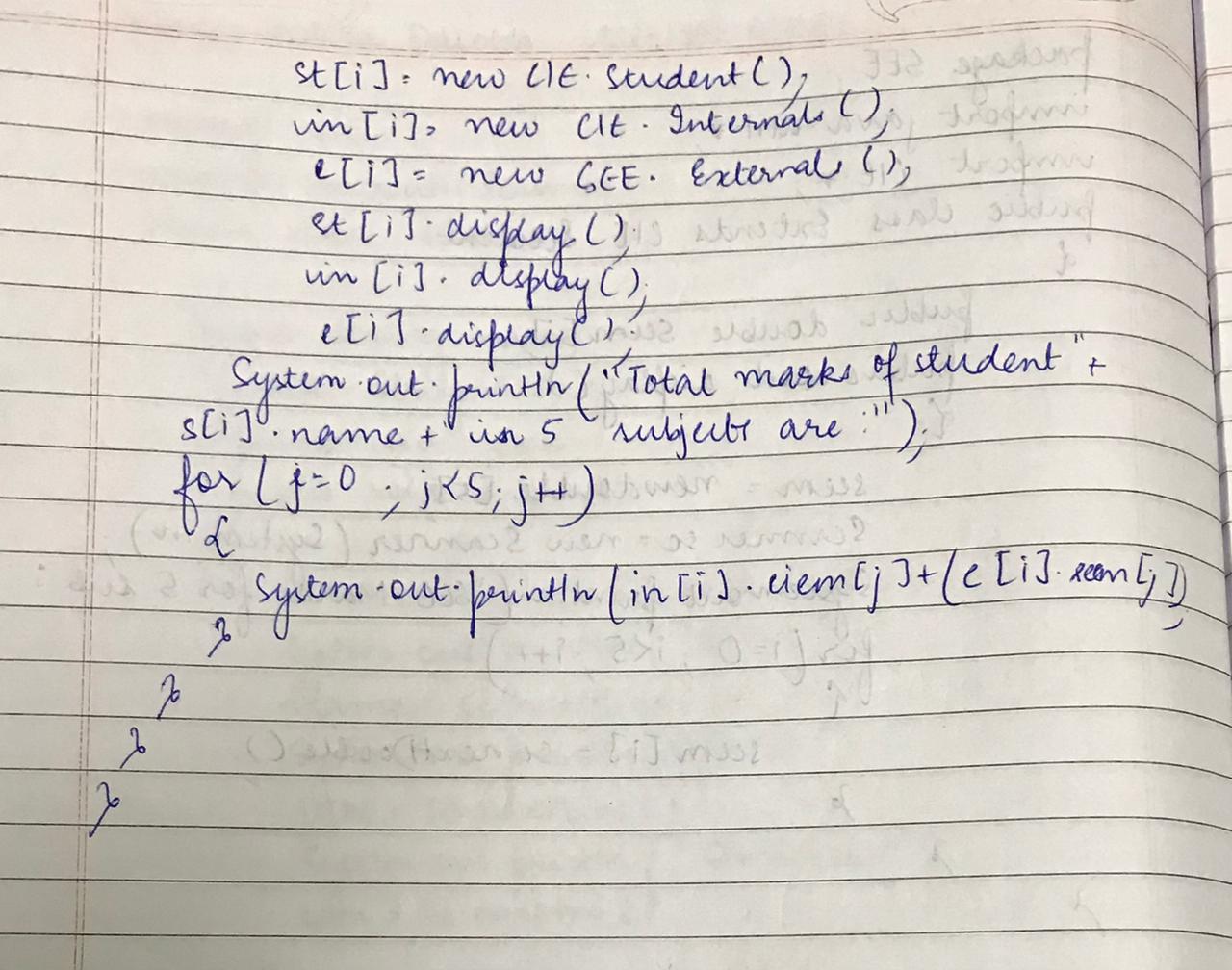
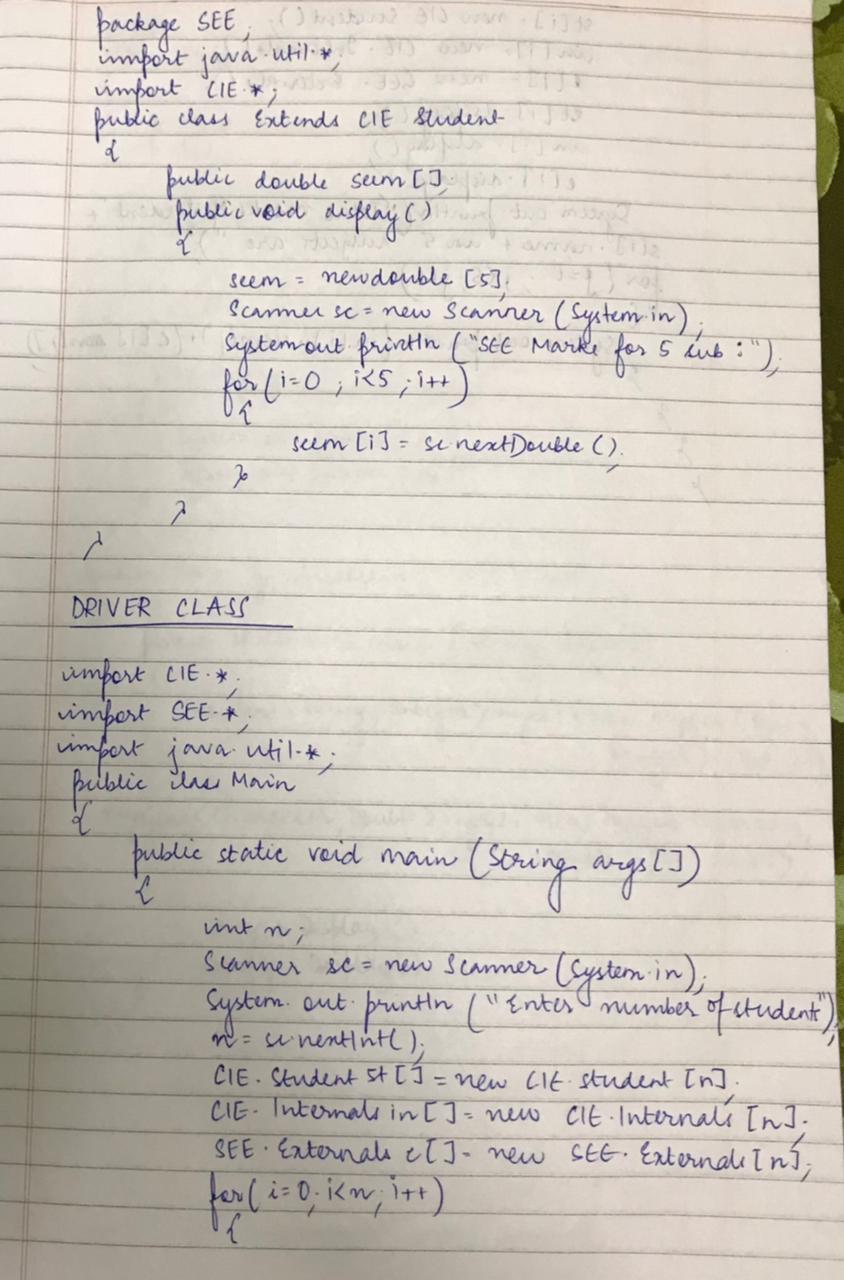
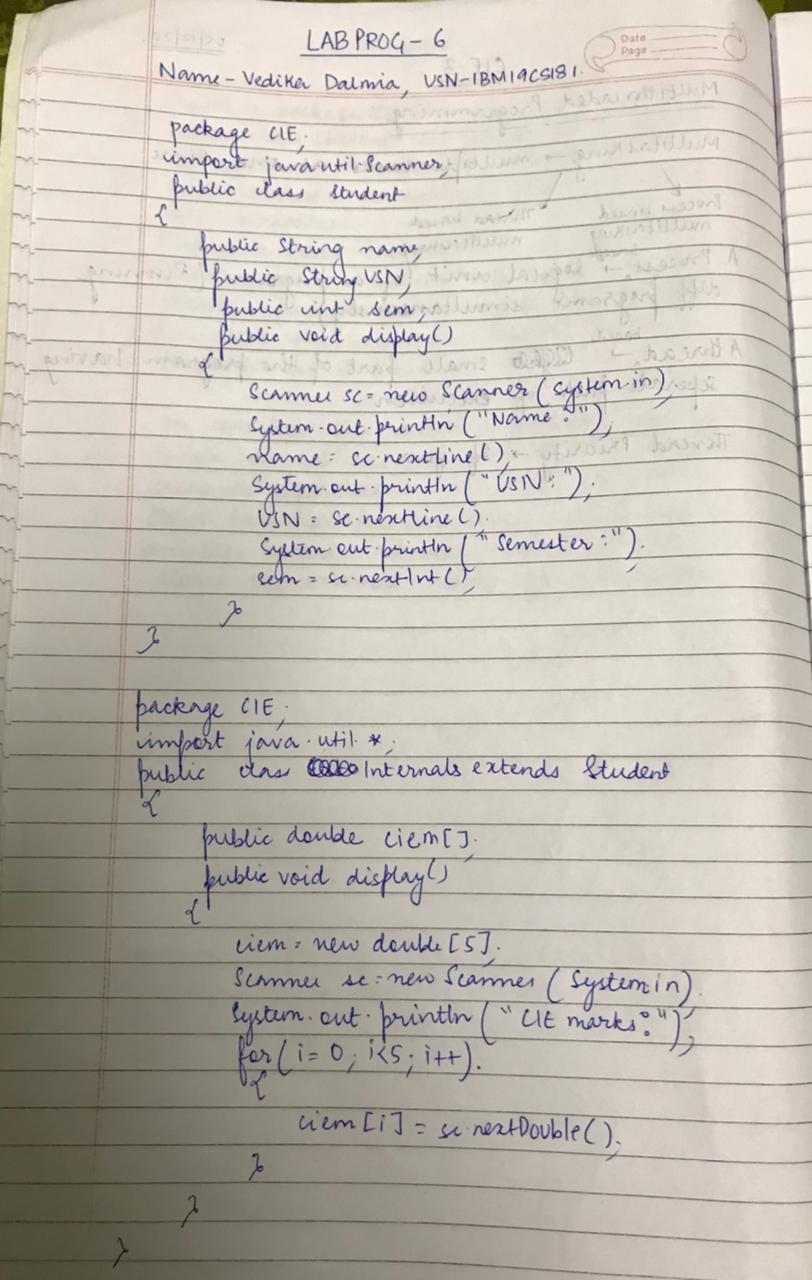
**OUTPUT SCREEN:5**



**Lab Program: 6**

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses. Below is two programs of class Student and class Internals. Both belong to a package named CIE. Class Internals extends from class Student.

**WRITEUP:**



**PROGRAM**

**File with name Student.java**

package CIE;

import java.util.Scanner;

public class Student

{

public String name;

public String usn;

public int sem;

public void display()

{

Scanner s=new Scanner(System.in);

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

name=s.next();

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

usn=s.next();

System.out.println("Semester:");

sem=s.nextInt();

}

}

**File with name Internals.java**

package CIE;

import java.util.Scanner;

public class Internals extends Student

{

public double ciem[];

public void display()

{

ciem=new double[5];

Scanner t=new Scanner(System.in);

System.out.println("CIE Marks for 5 subjects(out of 50):");

for(int i=0;i<5;i++)

ciem[i]=t.nextDouble();

}

}

**File with name Externals.java**

package SEE;

import CIE.\*;

import java.util.\*;

public class Externals extends CIE.Student

{

public double seem[];

public void display()

{

seem=new double[5];

Scanner s=new Scanner(System.in);

System.out.println("SEE Marks for 5 subjects(out of 100):");

for(int i=0;i<5;i++)

seem[i]=s.nextDouble();

}

}

**File with name Main.java**

import CIE.\*;

import SEE.\*;

import java.util.Scanner;

public class Main

{

public static void main(String args[])

{

int n;

Scanner s=new Scanner(System.in);

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

n=s.nextInt();

CIE.Student st[]=new CIE.Student[n];

CIE.Internals in[]=new CIE.Internals[n];

SEE.Externals e[]=new SEE.Externals[n];

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

{

st[i]=new CIE.Student();

in[i]=new CIE.Internals();

e[i]=new SEE.Externals();

st[i].display();

in[i].display();

e[i].display();

System.out.println("Total marks of student"+st[i].name+" in 5

subjects are:");

for(int j=0;j<5;j++)

{

System.out.println(in[i].ciem[j]+(e[i].seem[j]/2));

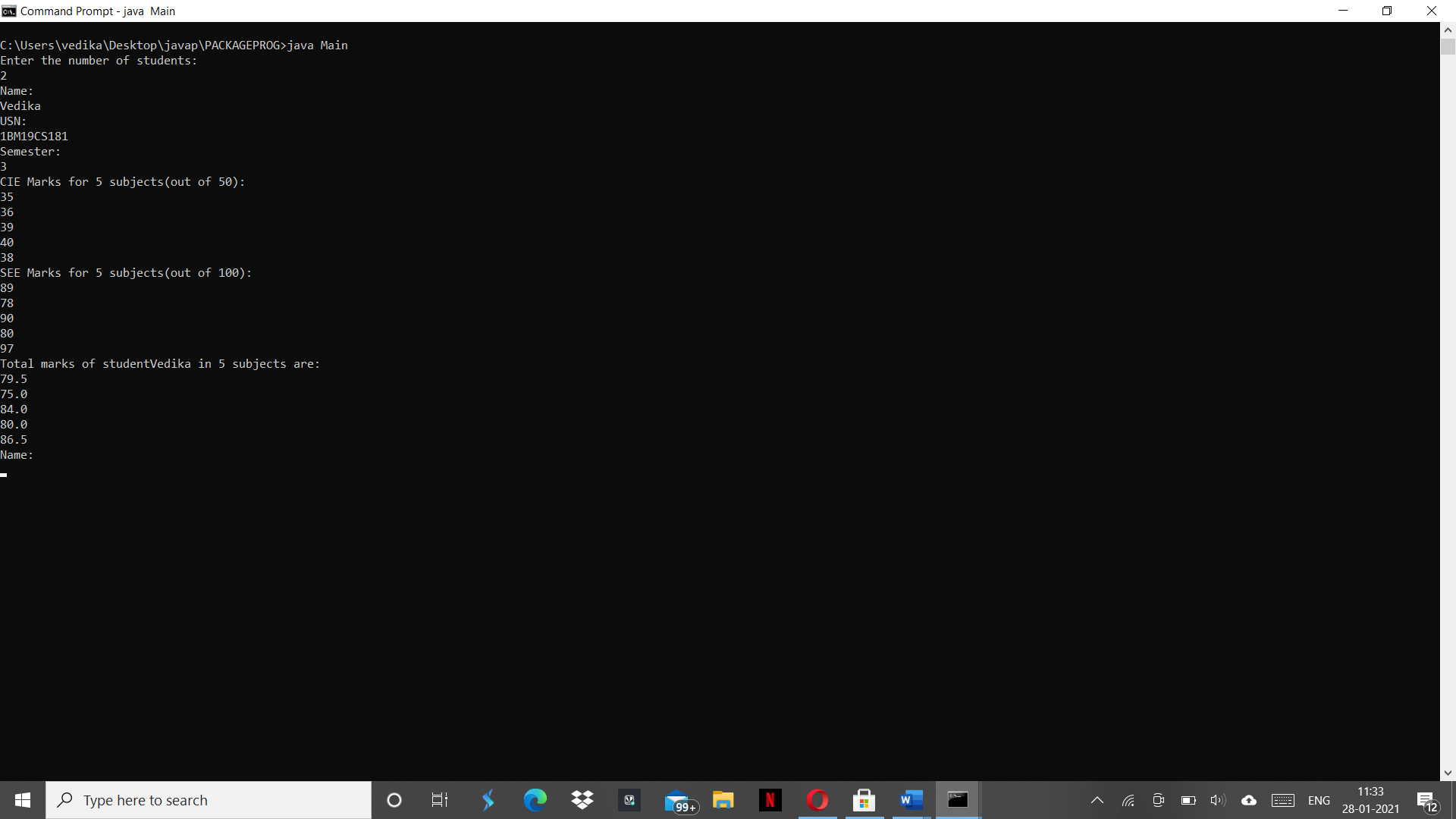
}

}

}

}

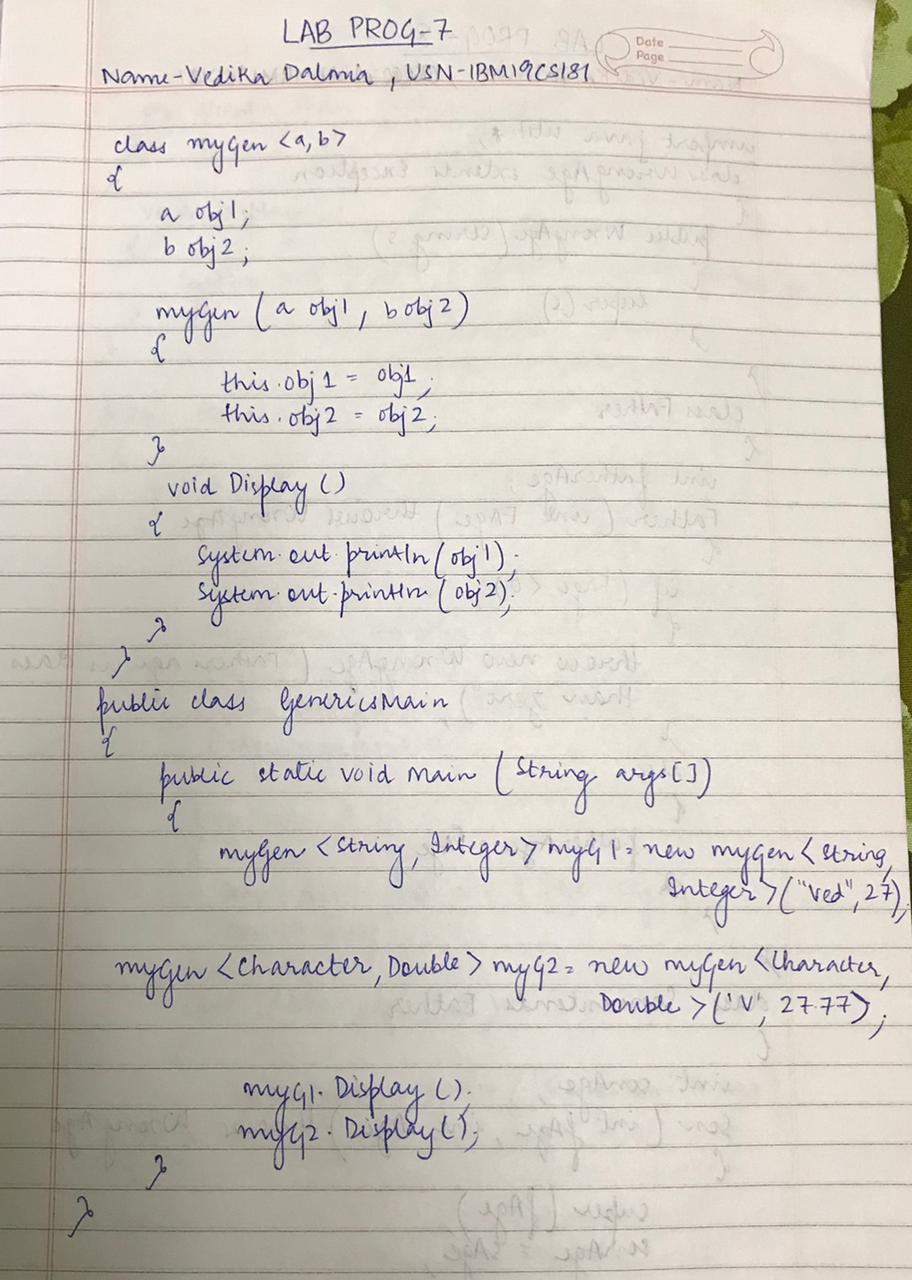
**OUTPUT SCREEN:6**



# Lab Program: 7

Write a program to demonstrate generics with multiple object parameters.

**WRITEUP:**



**PROGRAM**

class myGen<a,b>{

a obj1;

b obj2;

myGen(a obj1, b obj2){

this.obj1 = obj1;

this.obj2 = obj2;

}

void Display(){

System.out.println(obj1);

System.out.println(obj2);

}

}

public class Genericsmain{

public static void main(String args[]){

myGen<String,Integer>myG1 = new

myGen<String,Integer>("Vedika",27);

myGen<Character,Double>myG2 = new myGen<Character,Double>('V',27.77);

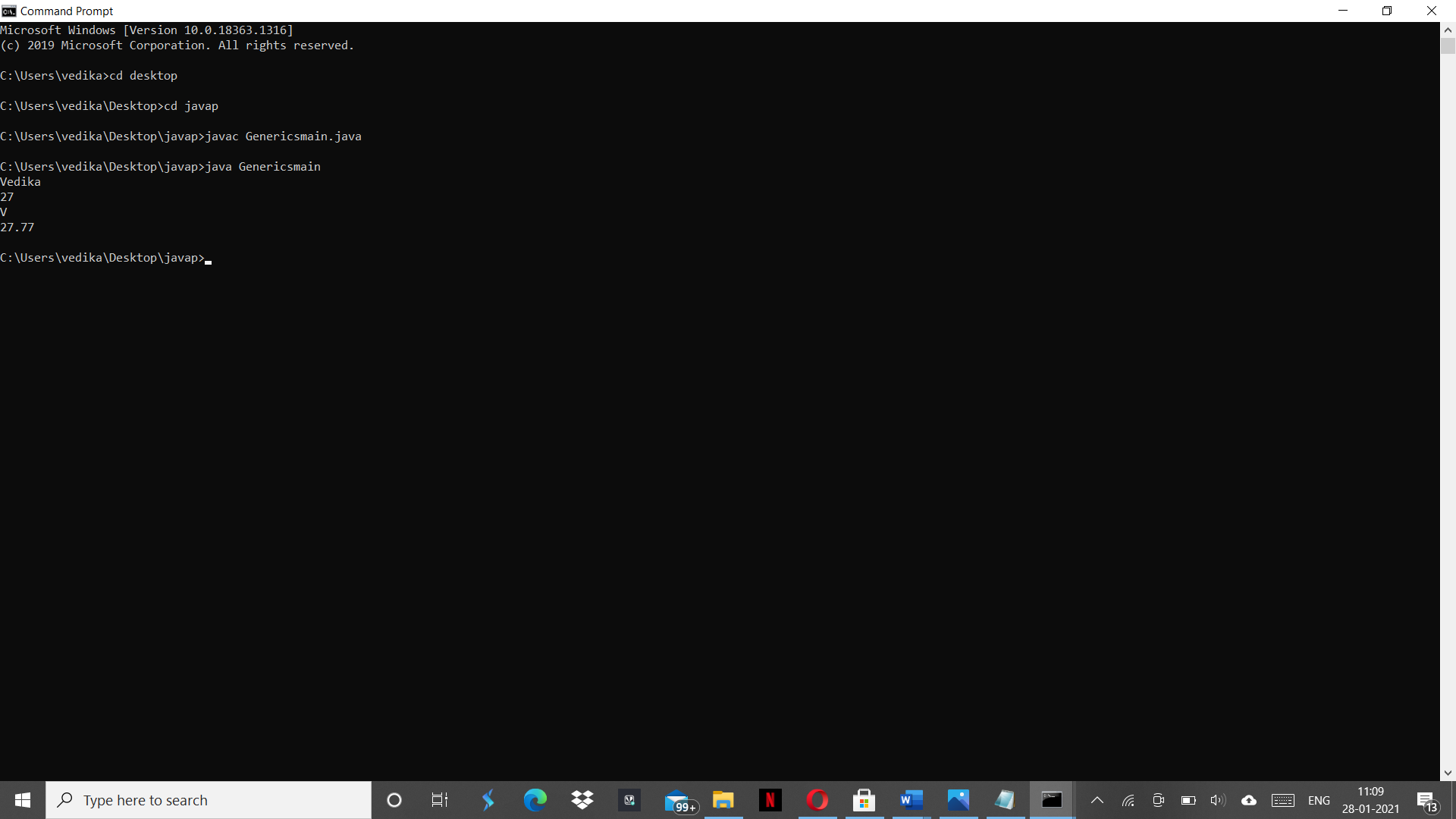
myG1.Display();

myG2.Display();

}

}

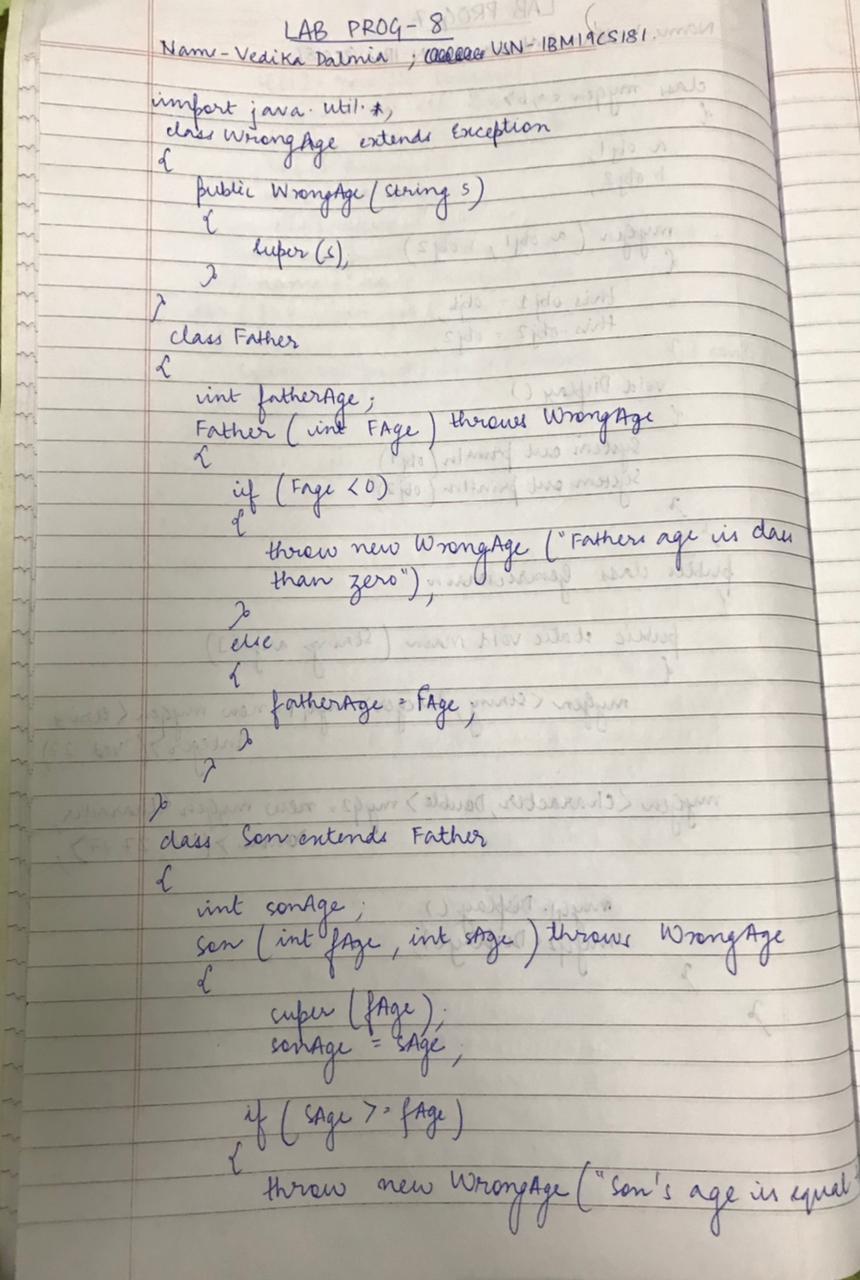
**OUTPUT SCREEN:7**

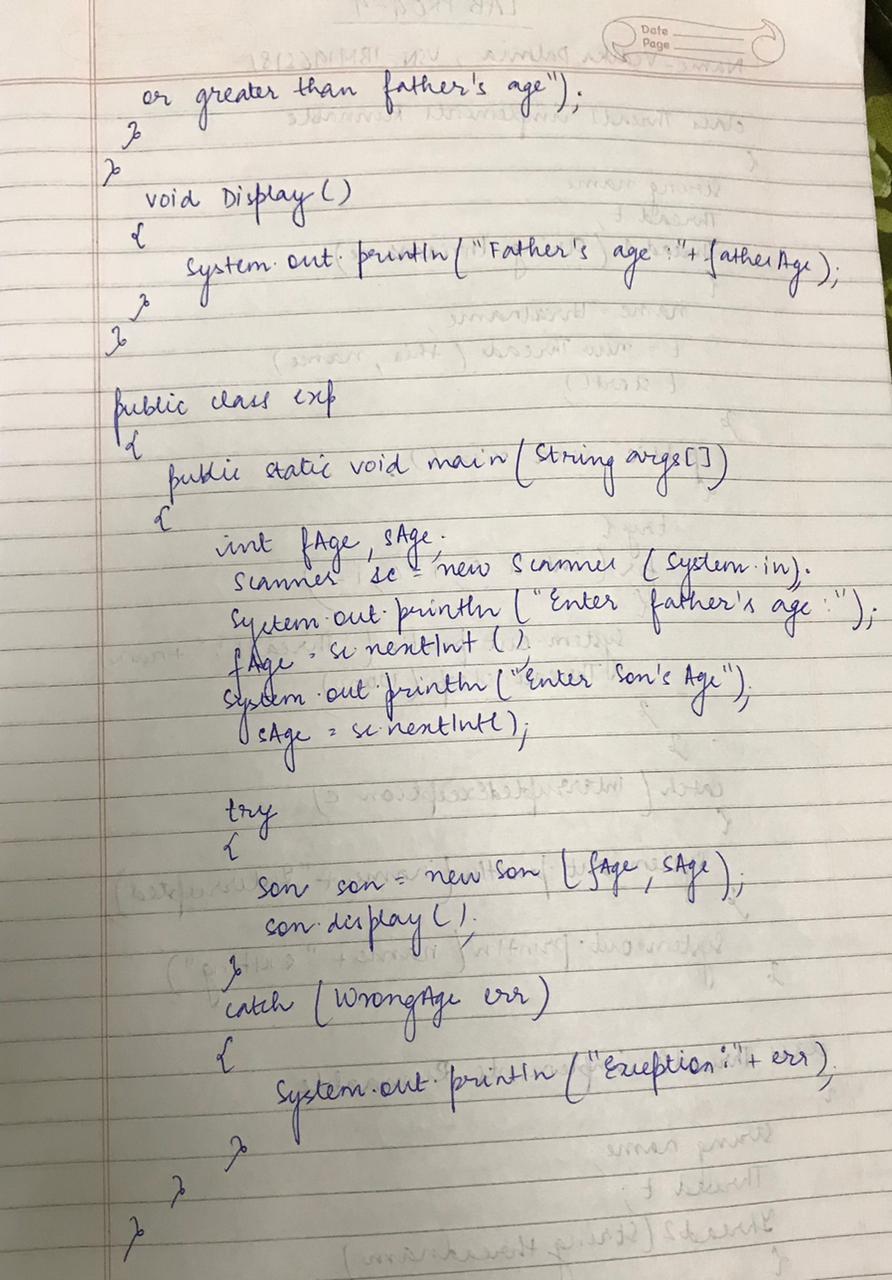


**Lab Program: 8**

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<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

**WRITEUP**





**PROGRAM**

import java.util.Scanner;

class WrongAge extends Exception

{

public WrongAge(String s)

{

super(s);

}

}

class Father

{

int fatherAge;

Father(int fAge) throws WrongAge{

if(fAge <=0)

{

throw new WrongAge("Father's age is less than 0");

}

else

{

this.fatherAge = fAge;

}

}

}

class Son extends Father

{

int sonAge;

Son(int fAge, int sAge) throws WrongAge{

super(fAge);

sonAge=sAge;

if(sAge >= fAge){

throw new WrongAge("Sons's age is equal to or greater than father's age");

}

}

void Display(){

System.out.println("Father's age: "+fatherAge);

System.out.println("Son's age: "+sonAge);

}

}

class lab8 {

public static void main(String[] args)

{

int fAge,sAge;

Scanner sc = new Scanner(System.in);

System.out.println("Enter father's age: ");

fAge = sc.nextInt();

System.out.println("Enter sons's age: ");

sAge = sc.nextInt();

try{

Son son = new Son(fAge, sAge);

son.Display();

}catch(WrongAge err)

{

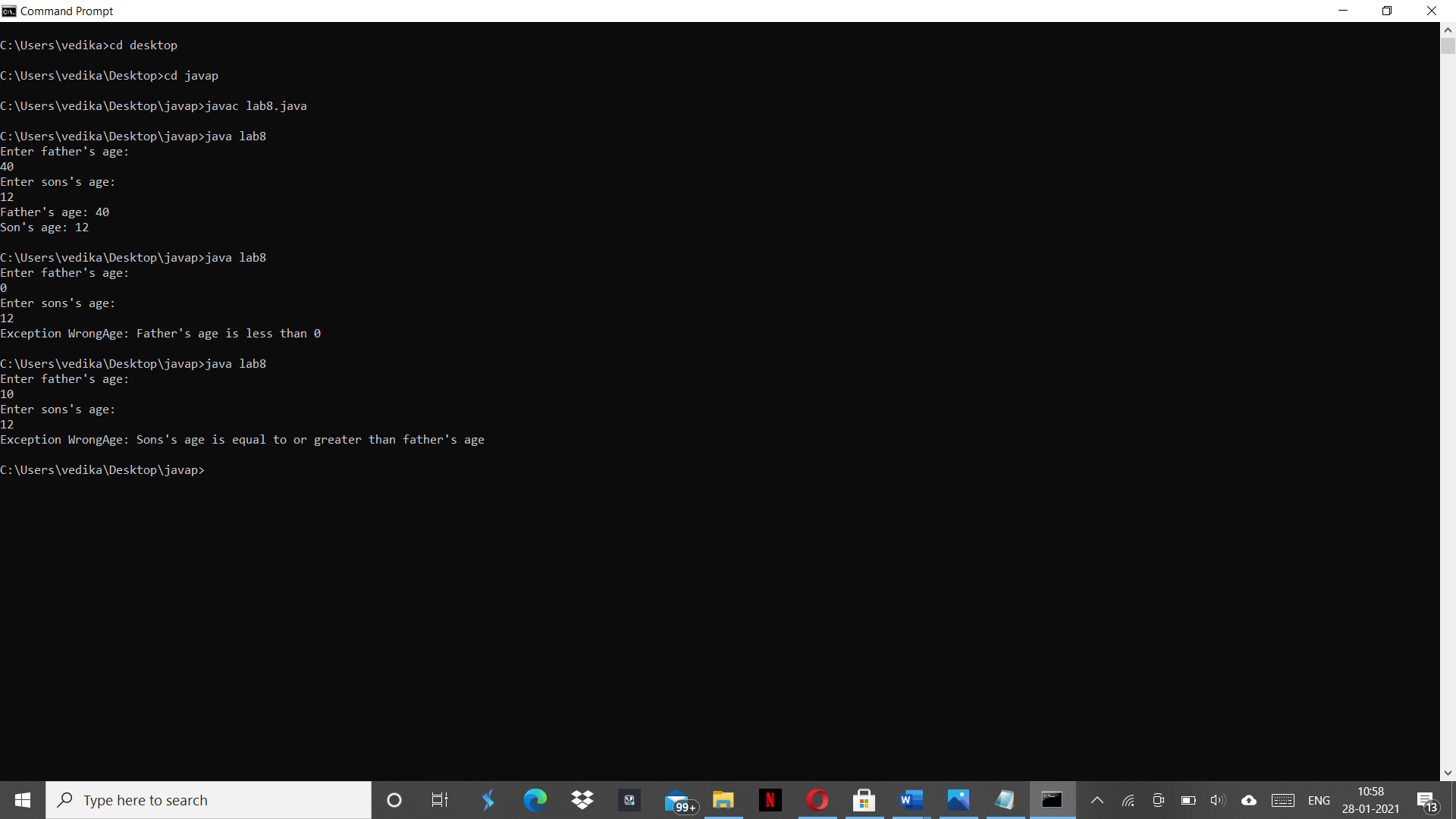
System.out.println("Exception " + err);

}

}

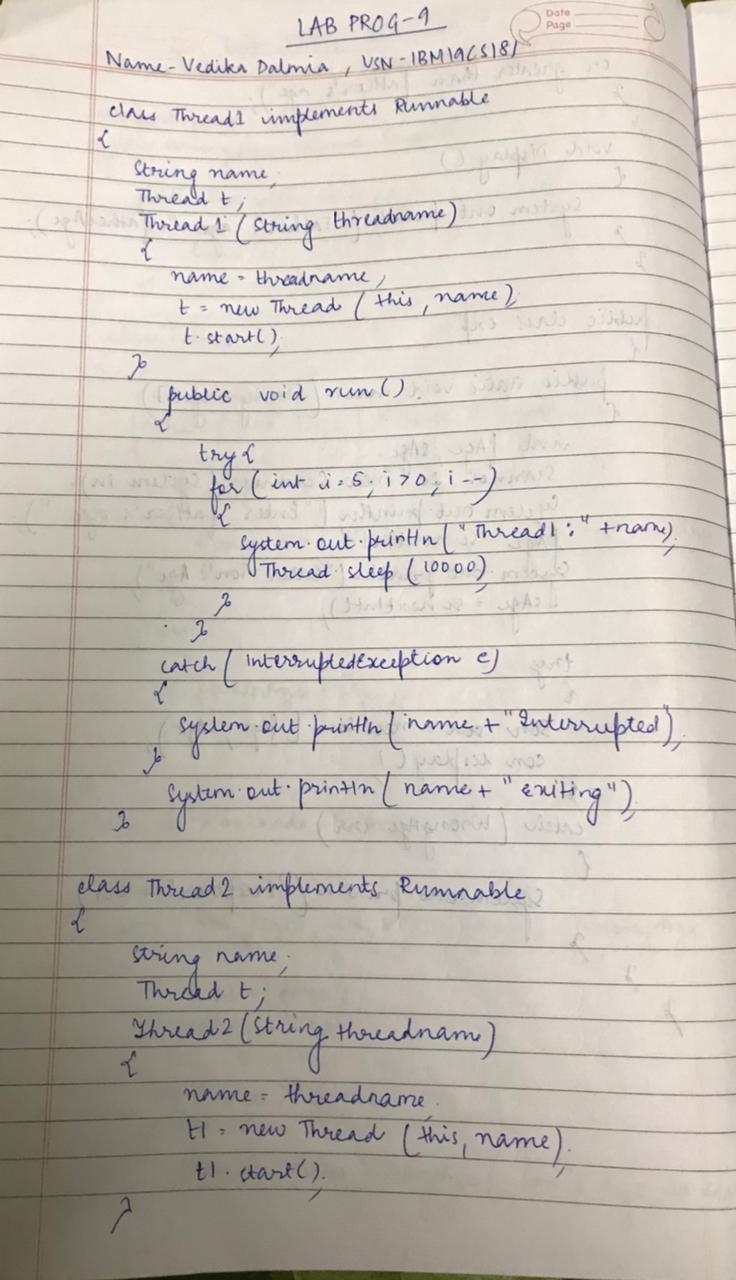
}

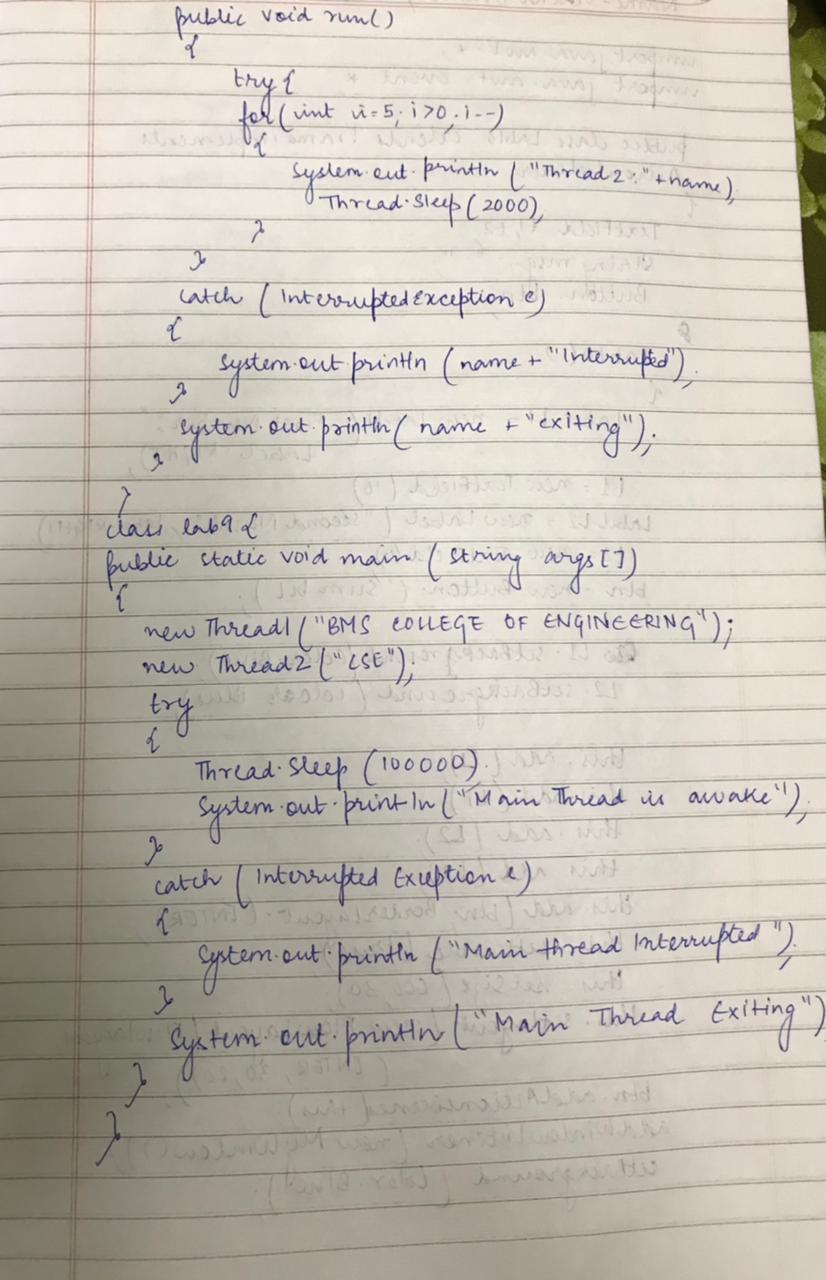
**OUTPUT SCREEN-8**



**Lab Program: 9**

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.

**WRITEUP**



**PROGRAM**

class Thread1 implements Runnable

{

String name;

Thread t;

Thread1(String threadname)

{

name=threadname;

t=new Thread(this,name);

t.start();

}

public void run()

{

try{

for(int i=5;i>0;i--)

{

System.out.println("Thread1"+name);

Thread.sleep(10000);

}

}

catch(InterruptedException e)

{

System.out.println(name+"Interrupted");

}

System.out.println(name+"exiting");

}

}

class Thread2 implements Runnable

{

String name;

Thread t1;

Thread2(String threadname)

{

name=threadname;

t1=new Thread(this,name);

t1.start();

}

public void run()

{

try{

for(int i=5;i>0;i--)

{

System.out.println("Thread2"+name);

Thread.sleep(2000);

}

}

catch(InterruptedException e)

{

System.out.println(name+"Interrupted");

}

System.out.println(name+"exiting");

}

}

class lab9{

public static void main(String args[])

{

new Thread1("BMS COLLEGE OF ENGINEERING");

new Thread2("CSE");

try{

Thread.sleep(100000);

System.out.println("Main thread is awake");

}

catch(InterruptedException e)

{

System.out.println("Main thread Interrupted");

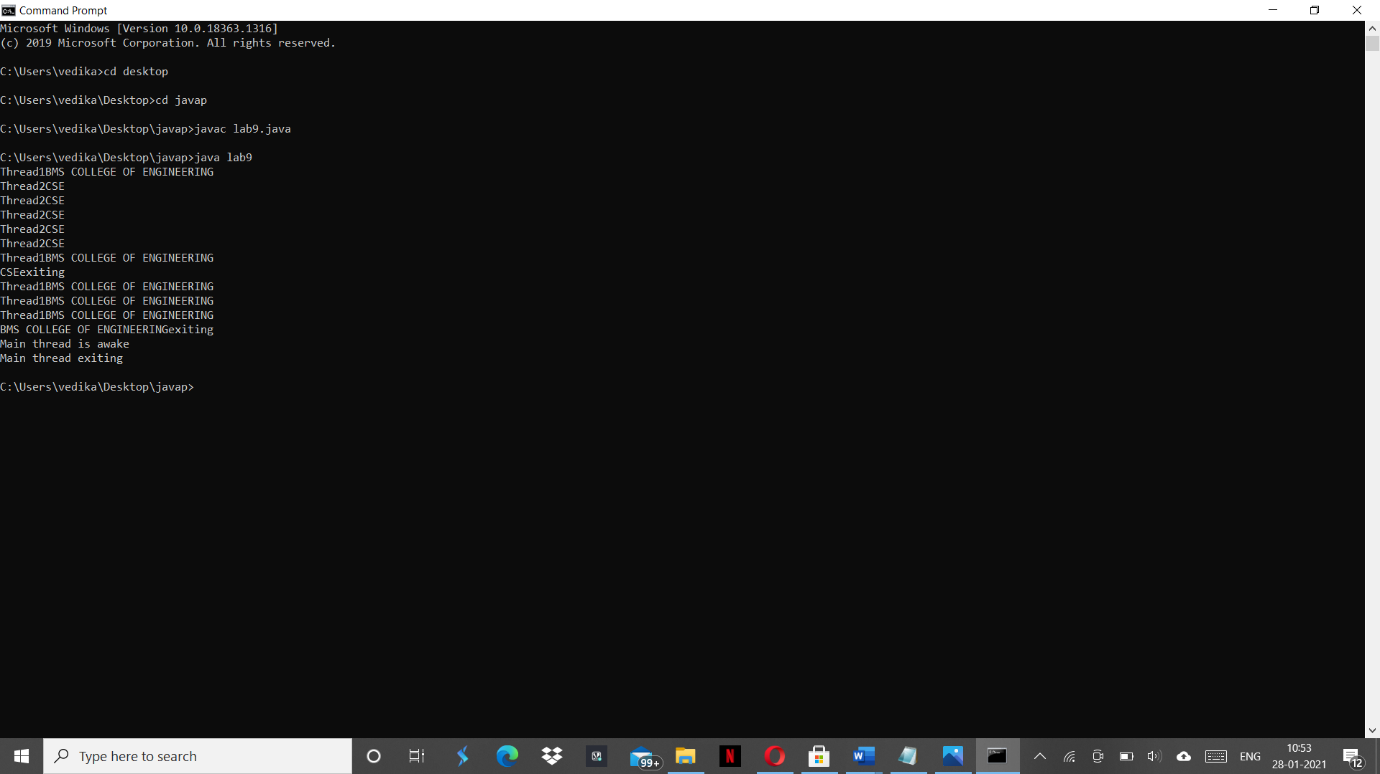
}

System.out.println("Main thread exiting");

}

}

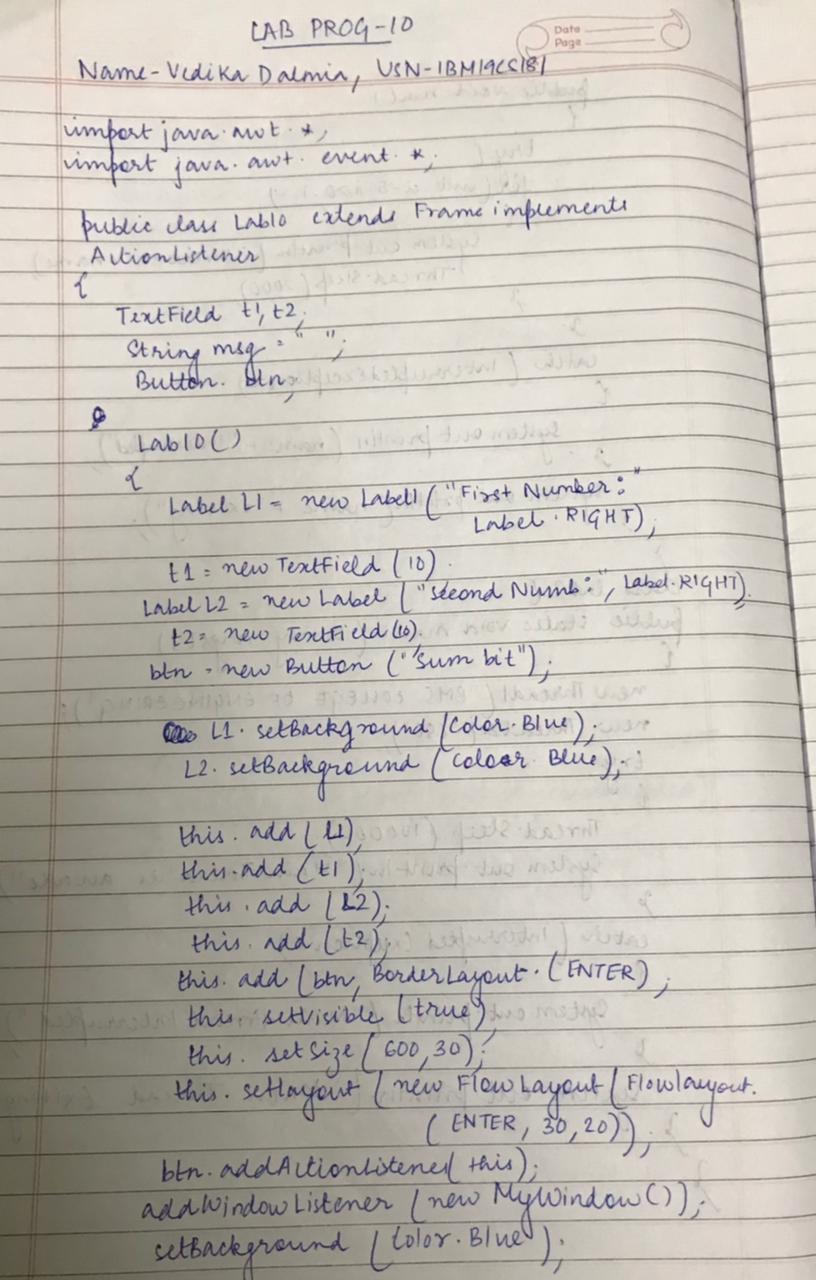
**OUTPUT SCREEN:9**

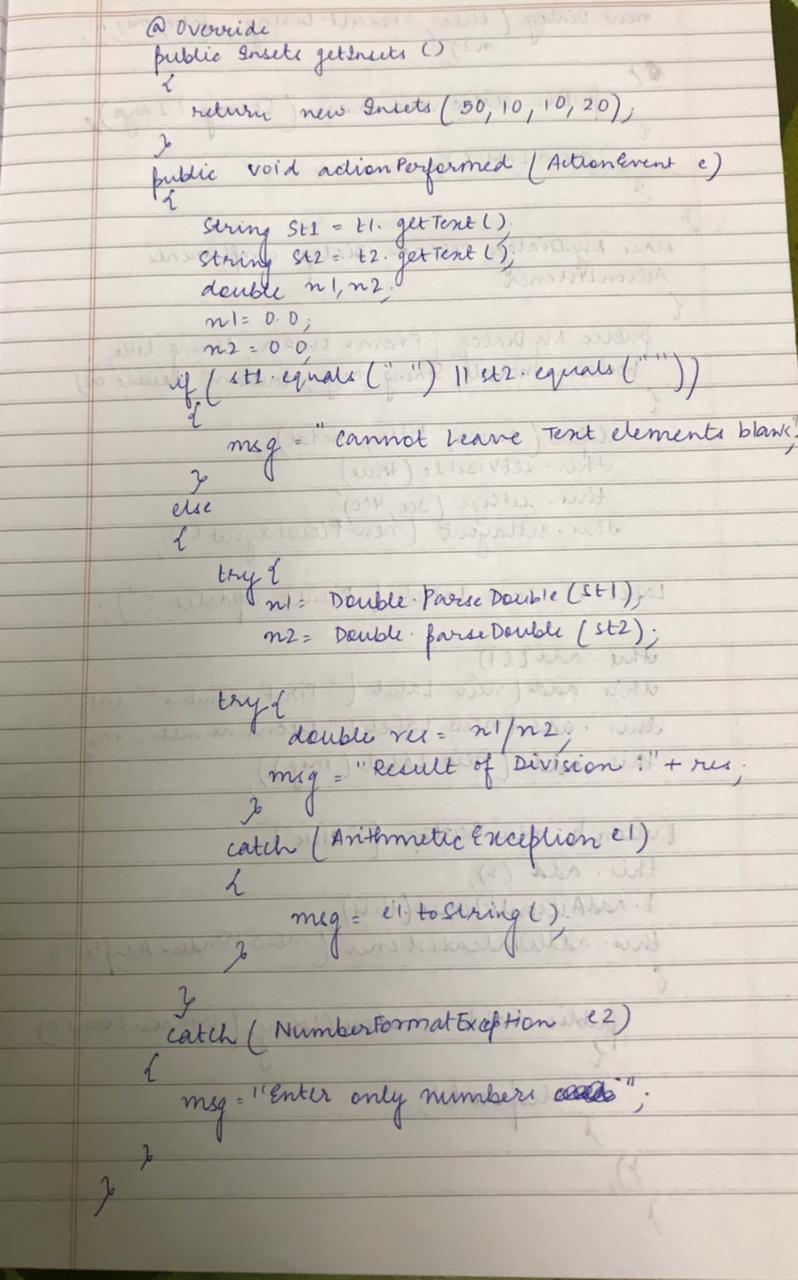


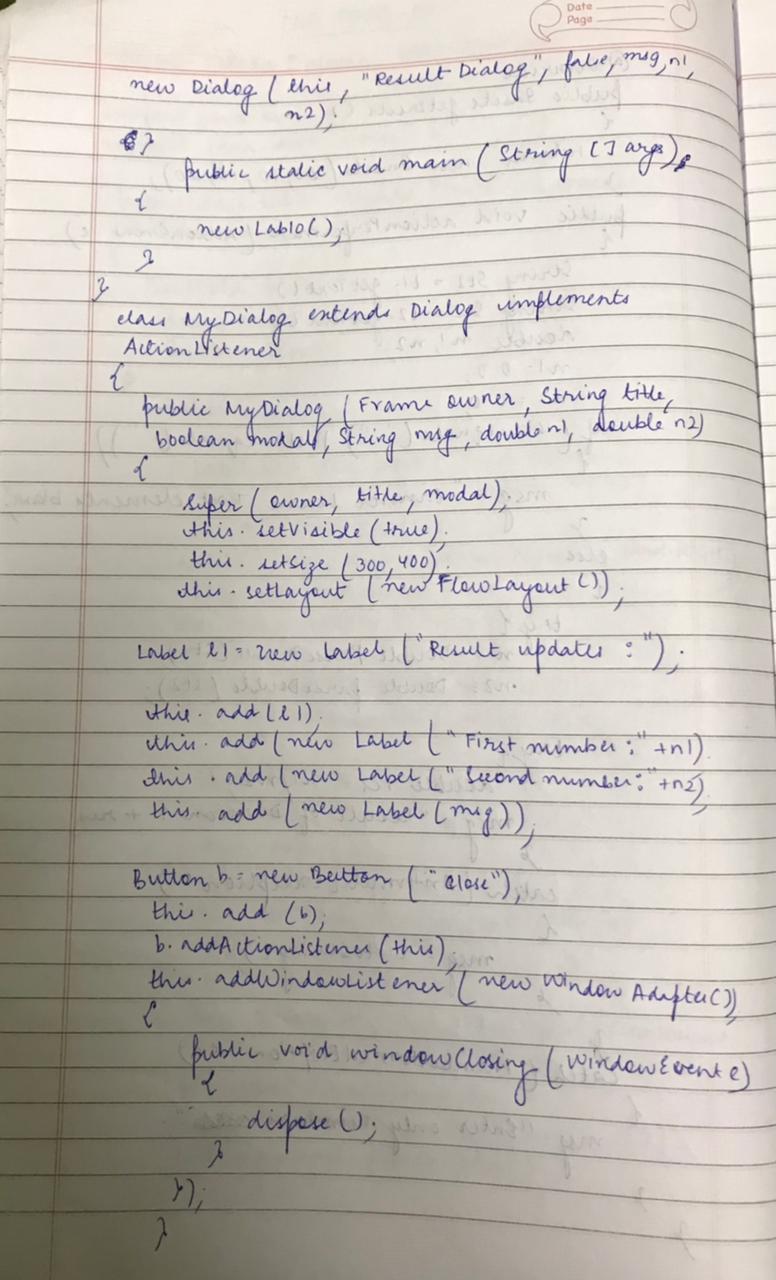
**Lab Program: 10**

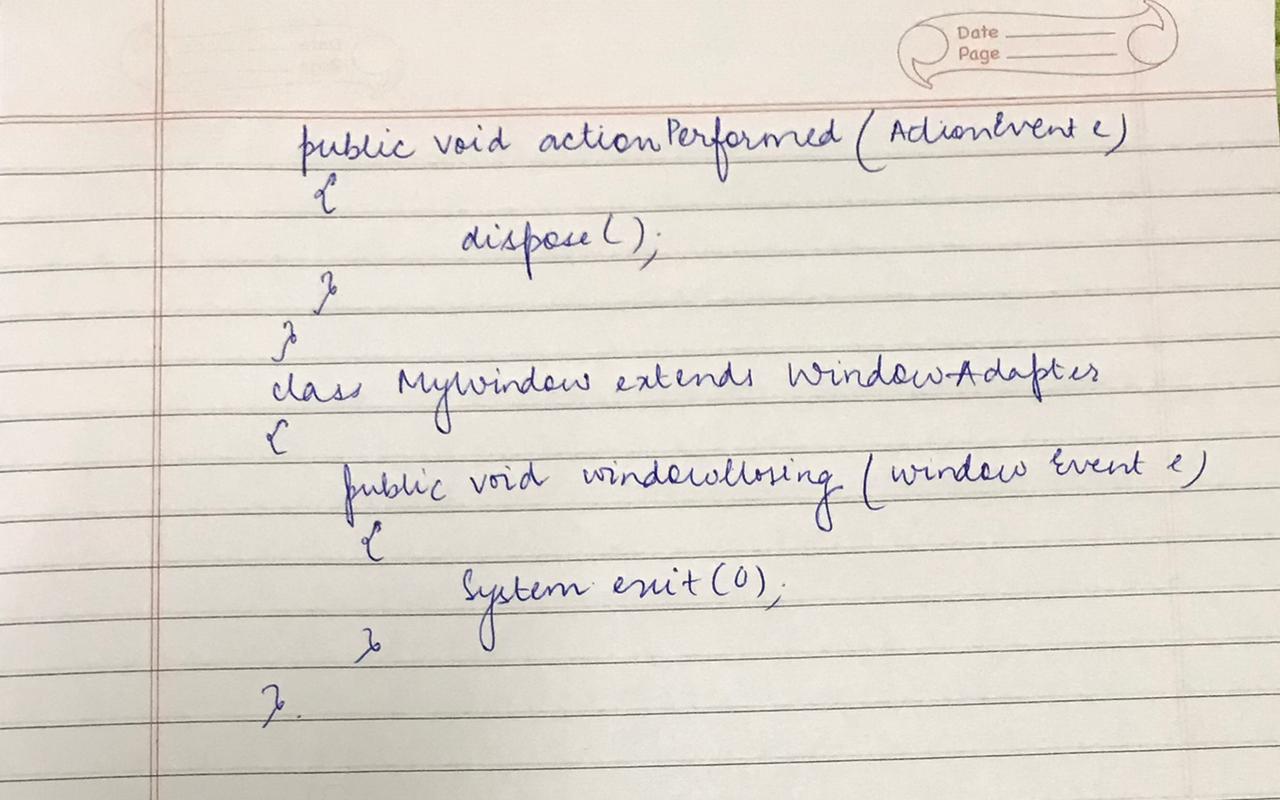
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

**WRITEUP**









**PROGRAM**

import java.awt.\*;

import java.awt.event.\*;

public class Lab10 extends Frame implements ActionListener{

TextField t1,t2;

String msg="";

Button btn;

Lab10(){

Label l1 = new Label("First Number: ",Label.RIGHT);

t1 = new TextField(10);

Label l2 = new Label("Second Number: ",Label.RIGHT);

t2 = new TextField(10);

btn = new Button("Submit");

//Label l = new Label("Updates:");

l1.setBackground(Color.BLUE);

l2.setBackground(Color.BLUE);

//this.setResizable(false);

this.add(l1);

this.add(t1);

this.add(l2);

this.add(t2);

//the following command will make sure that the input char is not visible to the user

//(it has been added just to demonstrate). Can be used for passwords.

//t1.setEchoChar('\*');

//t2.setEchoChar('#');

this.add(btn,BorderLayout.CENTER);

this.setVisible(true);

this.setSize(600, 300);

this.setLayout(new FlowLayout(FlowLayout.CENTER,20,10));

//t1.addActionListener(this);

btn.addActionListener(this);

addWindowListener(new MyWindow());

setBackground(Color.BLUE);

//System.out.println(BorderLayout.CENTER);

}

@Override

public Insets getInsets() {

return new Insets(50,10,10,20);

}

@Override

public void actionPerformed(ActionEvent e) {

String st1 = t1.getText();

String st2 = t2.getText();

double n1,n2;

n1 = 0.0;

n2 = 0.0;

if(st1.equals("")||st2.equals("")) {

msg="You cannot leave the text elements blank";

}else{

try {

n1 = Double.parseDouble(st1);

n2 = Double.parseDouble(st2);

try {

double res = n1/n2;

msg = "Result of division: "+res;

}catch(ArithmeticException e1) {

msg = e1.toString();

}

}catch(NumberFormatException e2) {

msg = "Enter only numbers and not other things";

}

}

new MyDialog(this,"Result Dialog",false,msg,n1,n2);

}

public static void main(String[] args) {

new Lab10();

}

}

class MyDialog extends Dialog implements ActionListener{

public MyDialog(Frame owner, String title, boolean modal,String msg, double n1,

double n2) {

super(owner, title, modal);

this.setVisible(true);

this.setSize(300, 400);

this.setLayout(new FlowLayout());

//System.out.println(owner);

Label l1 = new Label(" Updates on the result: ");

//l1.setSize(300, 20);

this.add(l1);

this.add(new Label("First Number: "+n1));

this.add(new Label("Second Number: "+n2));

this.add(new Label(msg));

Button b = new Button("Close");

this.add(b);

b.addActionListener(this);

this.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

dispose();

}

});

}

@Override

public void actionPerformed(ActionEvent e) {

dispose();

}

}

class MyWindow extends WindowAdapter{

public void windowClosing(WindowEvent e) {

System.exit(0);

}

}

**OUTPUT SCREEN:10**

