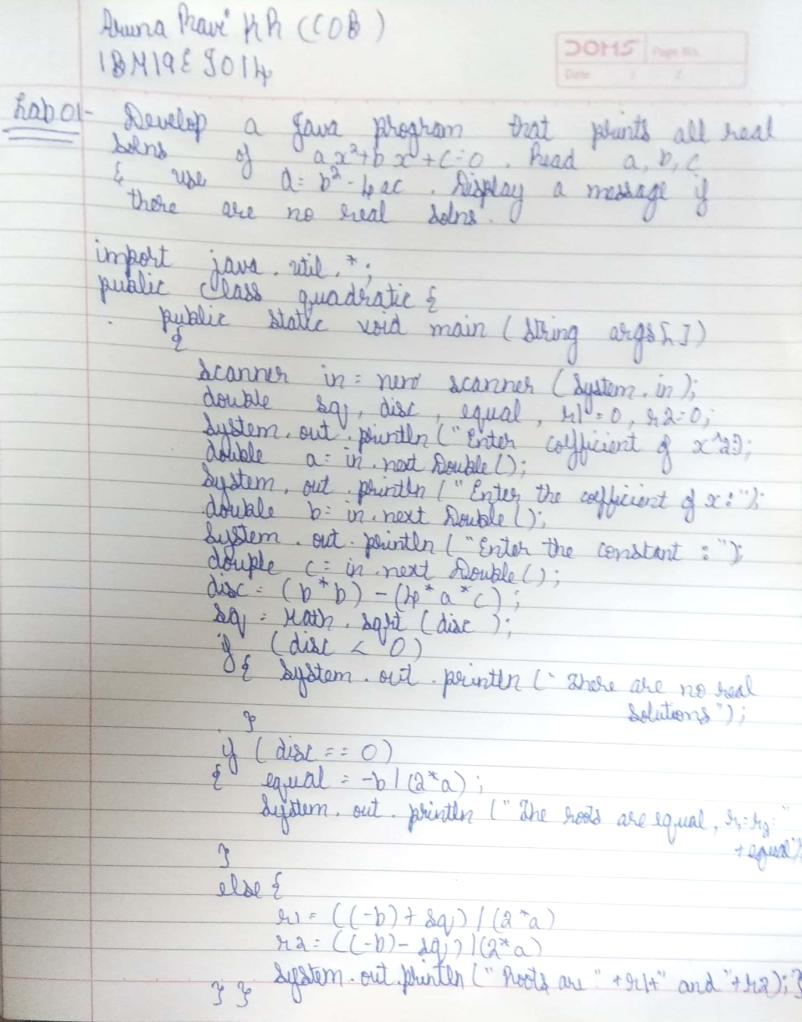
**Lab Program 1**

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.



**Source Code:**

import java.util.\*;

class javapg1

{

public static void main (String args[])

{

double a,b,c,d,r1,r2;

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

Scanner sc =new Scanner(System.in);

a=sc.nextDouble();

b=sc.nextDouble();

c=sc.nextDouble();

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

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

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

if (d>0)

{

System.out.println("root 1="+r1+"root 2="+r2);

}

else if (d ==0)

{

System.out.println("root 1=root2="+r1);

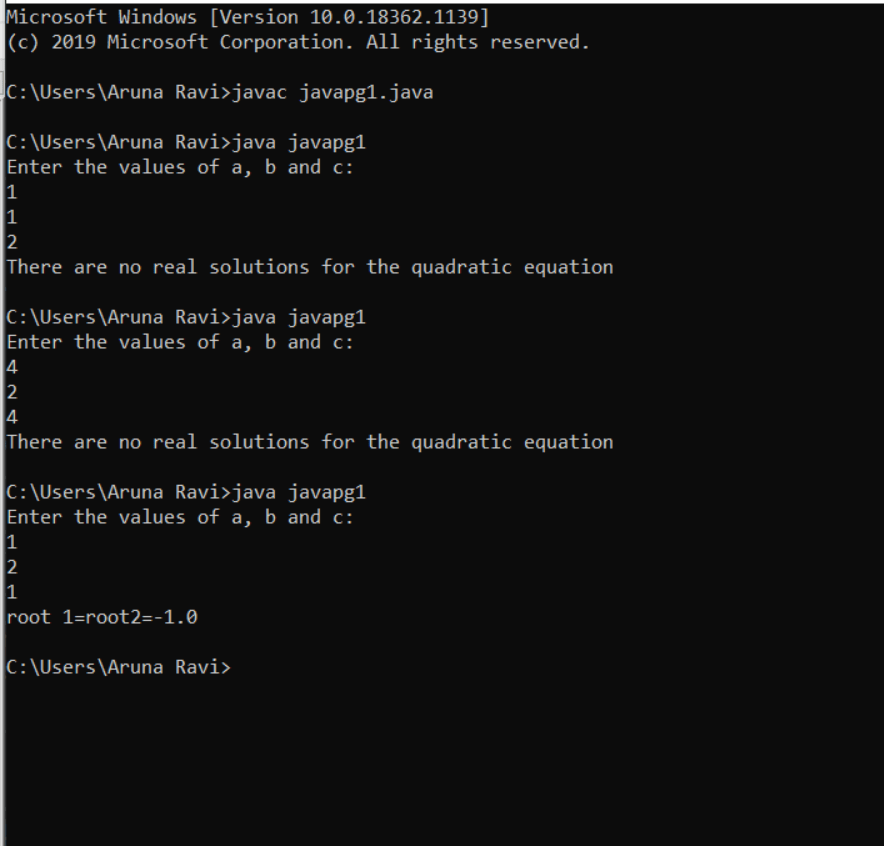
}

else

System.out.println("There are no real solutions for the quadratic equation");

}

}

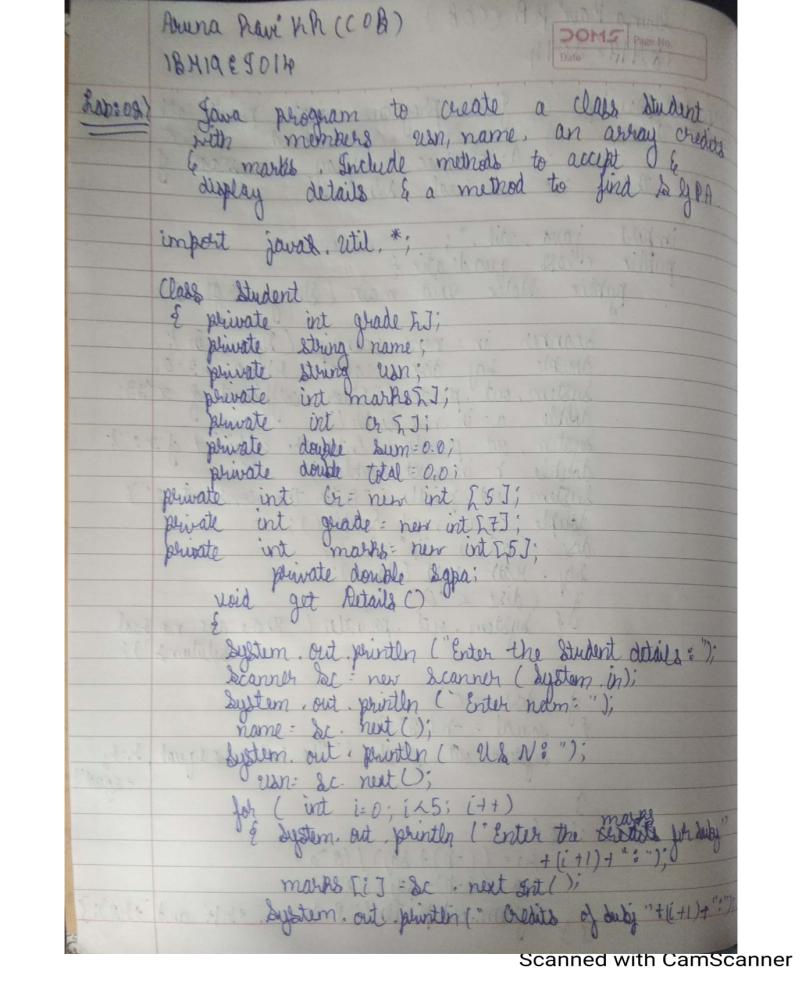


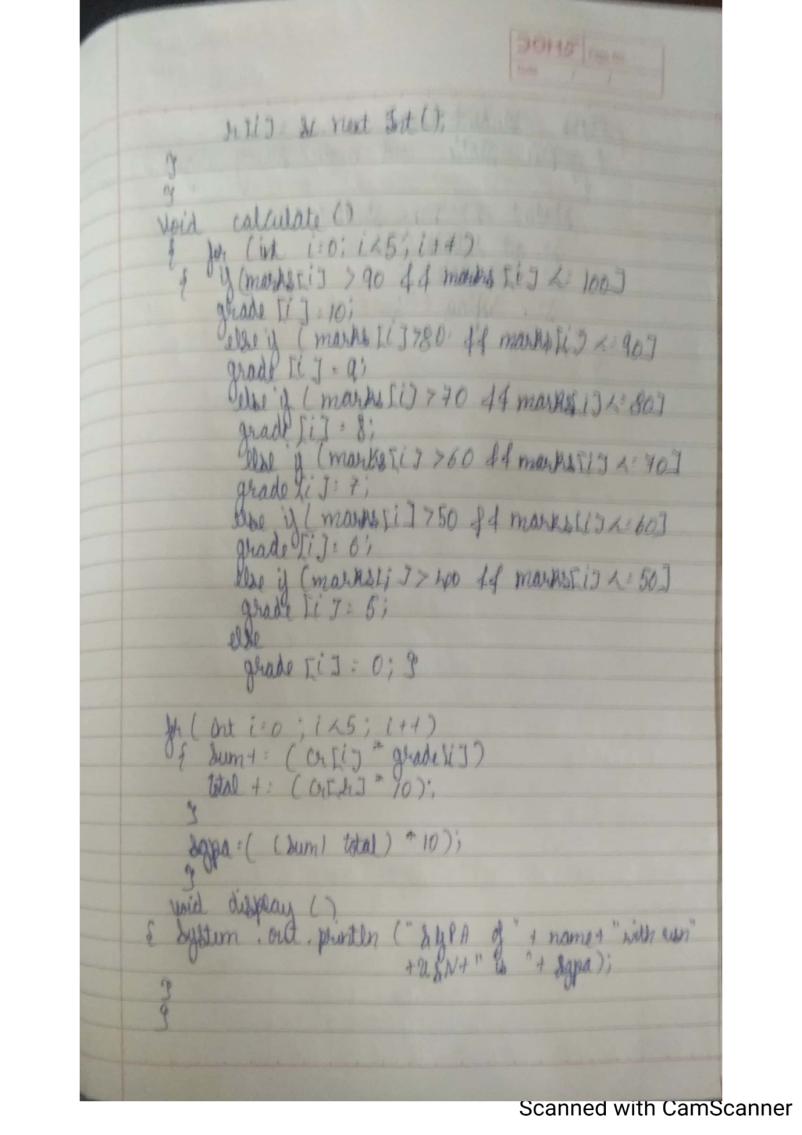
**Lab Program 2:**

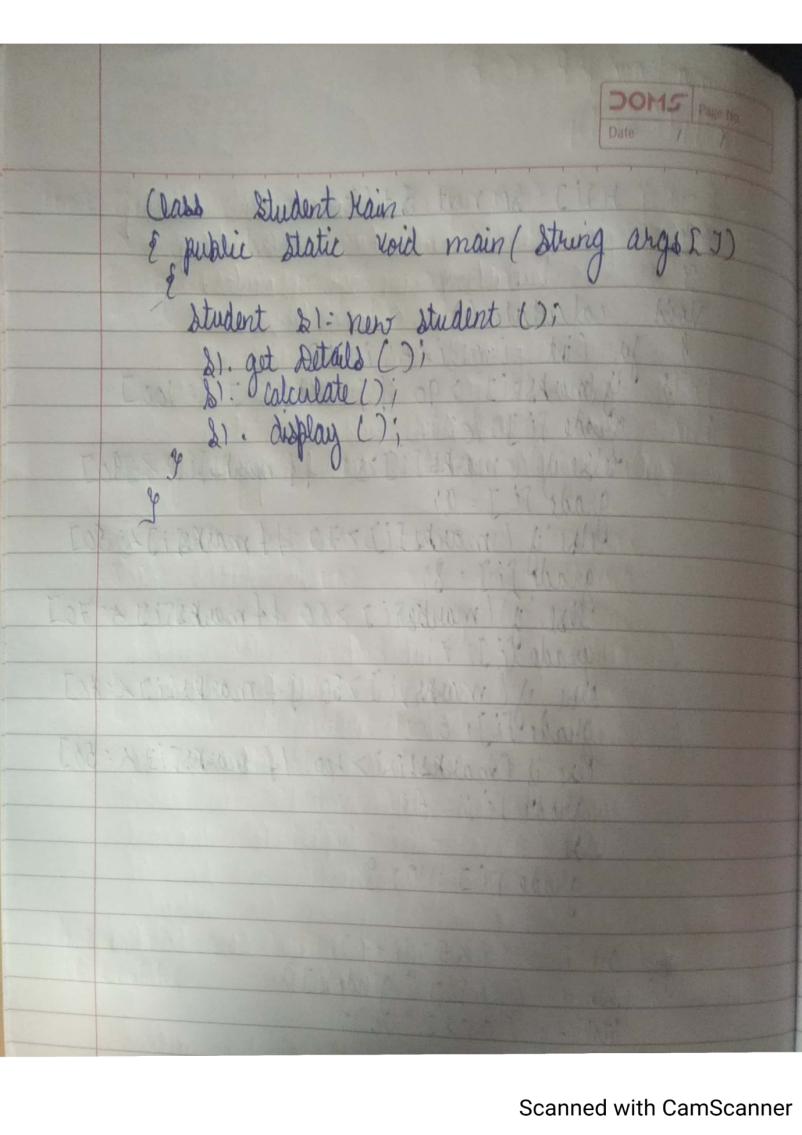
Develop a Java program to create a class Student with members usn, name, an array

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

calculate SGPA of a student.







**Source Code:**

import java.util.\*;

class Student

{

String name;

String USN;

int[] credits;

int[] marks;

Student(String yourName, String yourUSN, int[] yourCredits, int[] yourMarks )

{

this.name = yourName;

this.USN = yourUSN;

this.credits = yourCredits;

this.marks = yourMarks;

}

public int[] getGradePoints(int[] marksArray){

int[] gradePoints = new int[marksArray.length];

for(int i=0;i<marksArray.length;i++){

if (marksArray[i]>=90){

gradePoints[i] = 10;

}

else if (marksArray[i]>=80){

gradePoints[i] = 9;

}

else if (marksArray[i]>=70){

gradePoints[i] = 8;

}

else if (marksArray[i]>=60){

gradePoints[i] = 7;

}

else if (marksArray[i]>=50){

gradePoints[i] = 6;

}

else if (marksArray[i]>=40){

gradePoints[i] = 5;

}

else{

gradePoints[i] = 0;

}

}

return gradePoints;

}

public float calcArraySum(int[] anyArray){

float sum = 0;

for(int j=0;j<anyArray.length;j++){

sum += anyArray[j];

}

return sum;

}

public float calcCGPA(int[] creditsArray, int[] marksArray){

int[] totalGradePoints = new int[marksArray.length];

int[] gradePointsArray = getGradePoints(marksArray);

for(int k=0;k<marksArray.length;k++){

totalGradePoints[k] = creditsArray[k] \* gradePointsArray[k];

}

float sumGradePoints = calcArraySum(totalGradePoints);

float totalCredits = calcArraySum(creditsArray);

float CGPA = sumGradePoints/totalCredits;

return CGPA;

}

public void dispDetails(){

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

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

System.out.println("Credits: " + Arrays.toString(this.credits));

System.out.println("Marks: " + Arrays.toString(this.marks));

System.out.println("CGPA: " + calcCGPA(this.credits, this.marks));

}

}

class SGPA{

public static int[] getArrayFromString(String stringedArray){

String[] stringValues = stringedArray.split(",");

int[] arrayFromString = new int[stringValues.length];

for(int i=0;i<stringValues.length;i++){

arrayFromString[i] = Integer.parseInt(stringValues[i]);

}

return arrayFromString;

}

public static void main(String[] args){

Scanner getData = new Scanner(System.in);

System.out.println("Enter the number of Students and number of Subjects: ");

int numStudents = getData.nextInt();

int numSubjects = getData.nextInt();

Student studentList[] = new Student[numStudents];

Scanner getDetails = new Scanner(System.in);

for(int k=0;k<numStudents;k++){

System.out.println("Enter the details of student " + k + ": ");

studentList[k] = new Student(getDetails.next(),getDetails.next(),getArrayFromString(getDetails.next()),getArrayFromString(getDetails.next()));

}

System.out.println("\n\*\*\*\*\* STUDENT DETAILS with CGPA \*\*\*\*\*\n");

for(int h=0;h<studentList.length;h++){

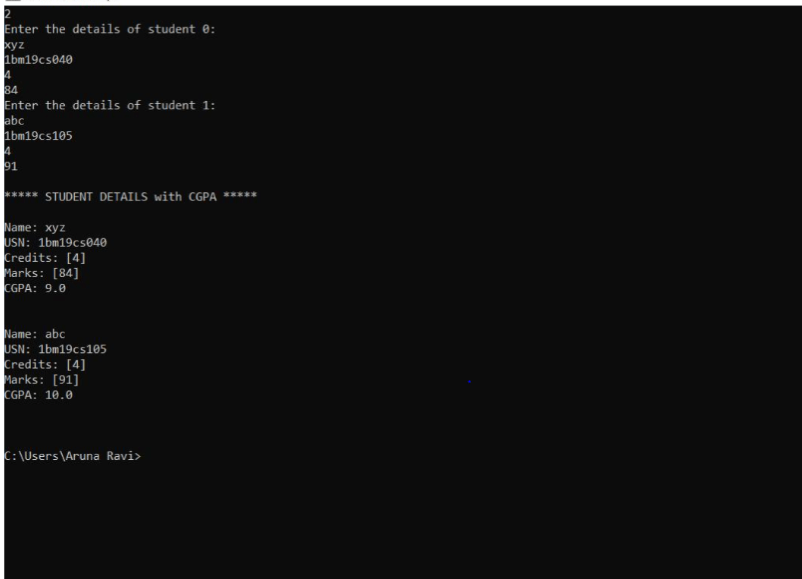
studentList[h].dispDetails();

System.out.println("\n");

}

}

}



**Lab Program - 3**

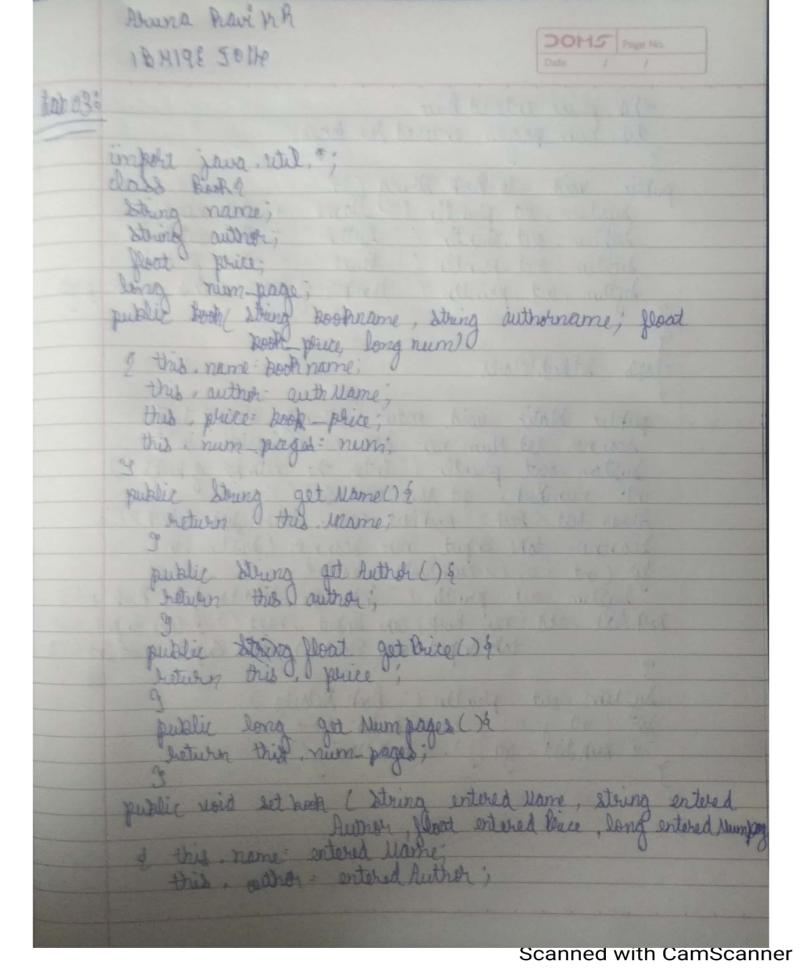
Create a class Book which contains four members: name, author, price,

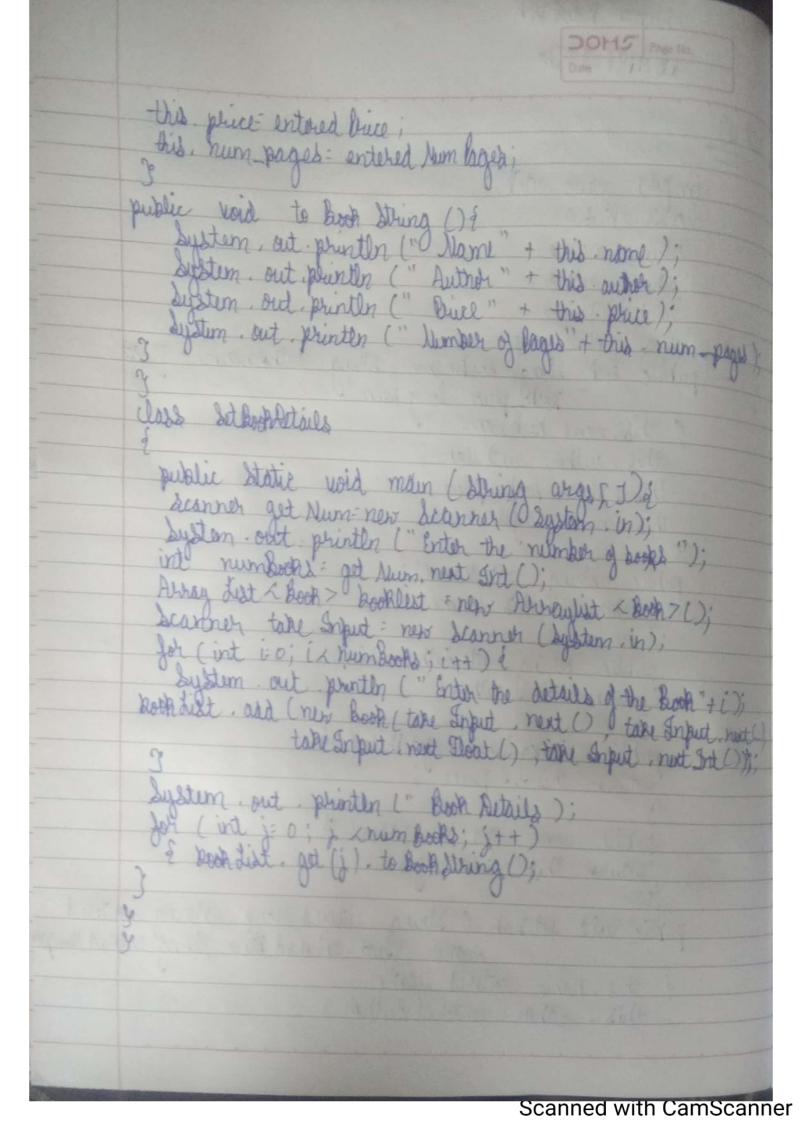
num\_pages. Include a constructor to set the values for the members. Include

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.





**Source Code:**

import java.util.\*;

class Book{

String name;

String author;

float price;

long num\_pages;

public Book(String bookName, String authName, float book\_price, long num){

this.name = bookName;

this.author = authName;

this.price = book\_price;

this.num\_pages = num;

}

public String getName(){

return this.name;

}

public String getAuthor(){

return this.author;

}

public float getPrice(){

return this.price

}

public long getNumPages(){

return this.num\_pages;

}

public void setBook(String enteredName, String enteredAuthor, float enteredPrice, long enteredNumPages){

this.name = enteredName;

this.author = enteredAuthor;

this.price = enteredPrice;

this.num\_pages = enteredNumPages;

}

public void toBookString(){

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

System.out.println("Author : " + this.author);

System.out.println("Price : " + this.price);

System.out.println("Number of Pages : " + this.num\_pages);

System.out.println("\n");

}

}

class SetBookDetails{

public static void main(String[] args){

Scanner getNum = new Scanner(System.in);

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

int numBooks = getNum.nextInt();

ArrayList<Book> bookList = new ArrayList<Book>();

Scanner takeInput = new Scanner(System.in);

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

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

bookList.add(new Book(takeInput.next(),takeInput.next(),takeInput.nextFloat(),takeInput.nextInt()));

}

System.out.println("\n\*\*\*\*\* BOOK DETAILS \*\*\*\*\*\n");

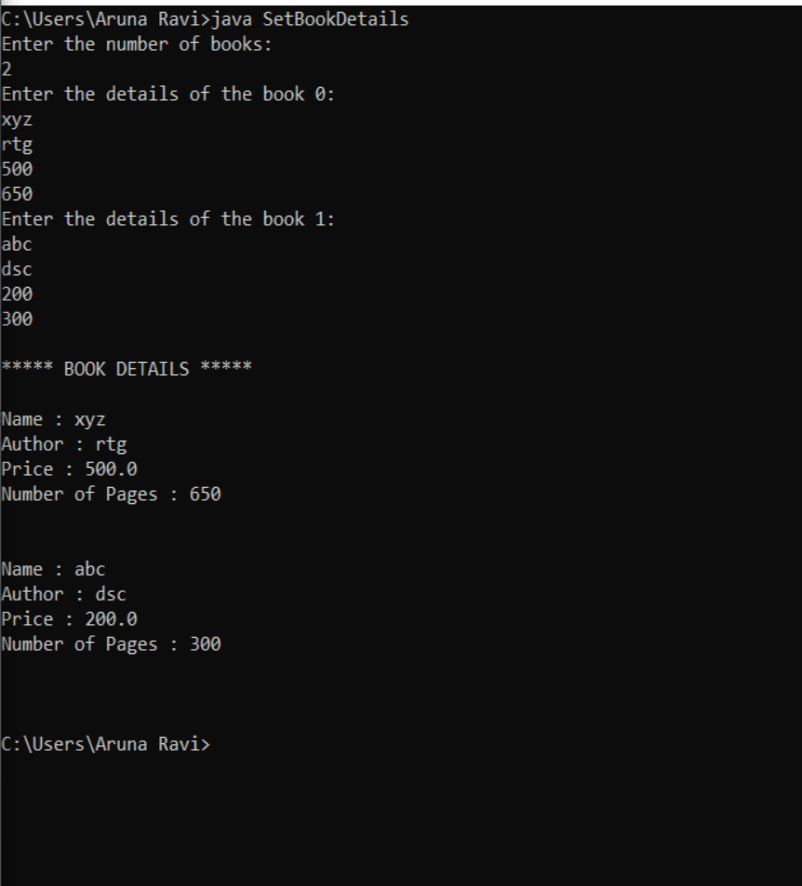
for(int j=0;j<numBooks;j++){

bookList.get(j).toBookString();

}

}

}



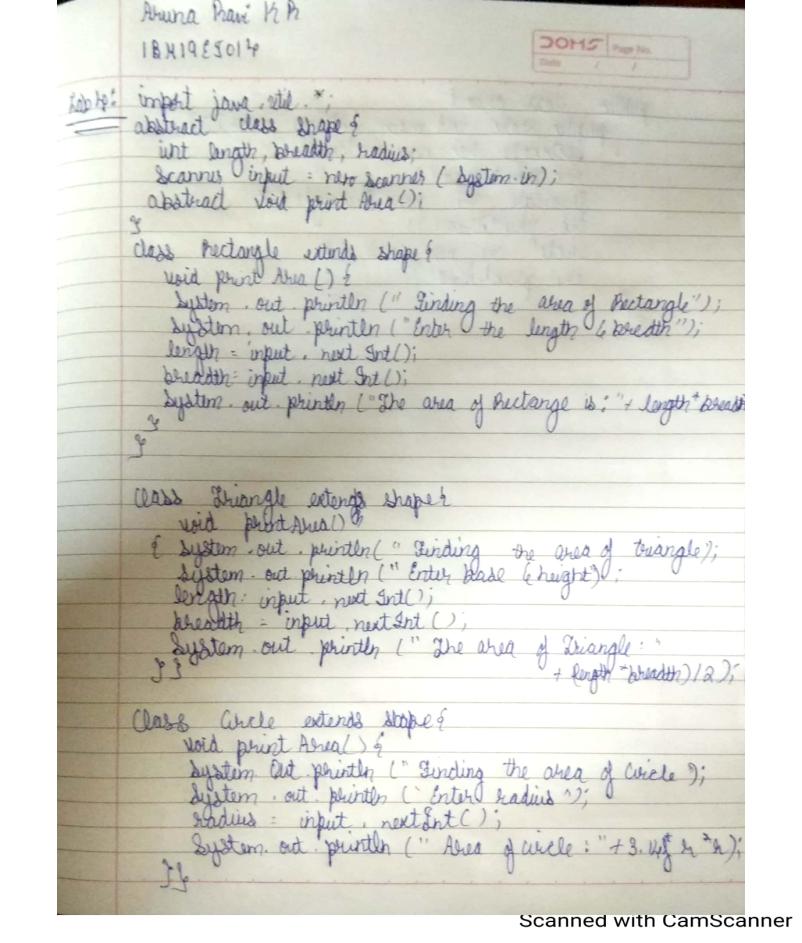
**Lab Exercises – 4**

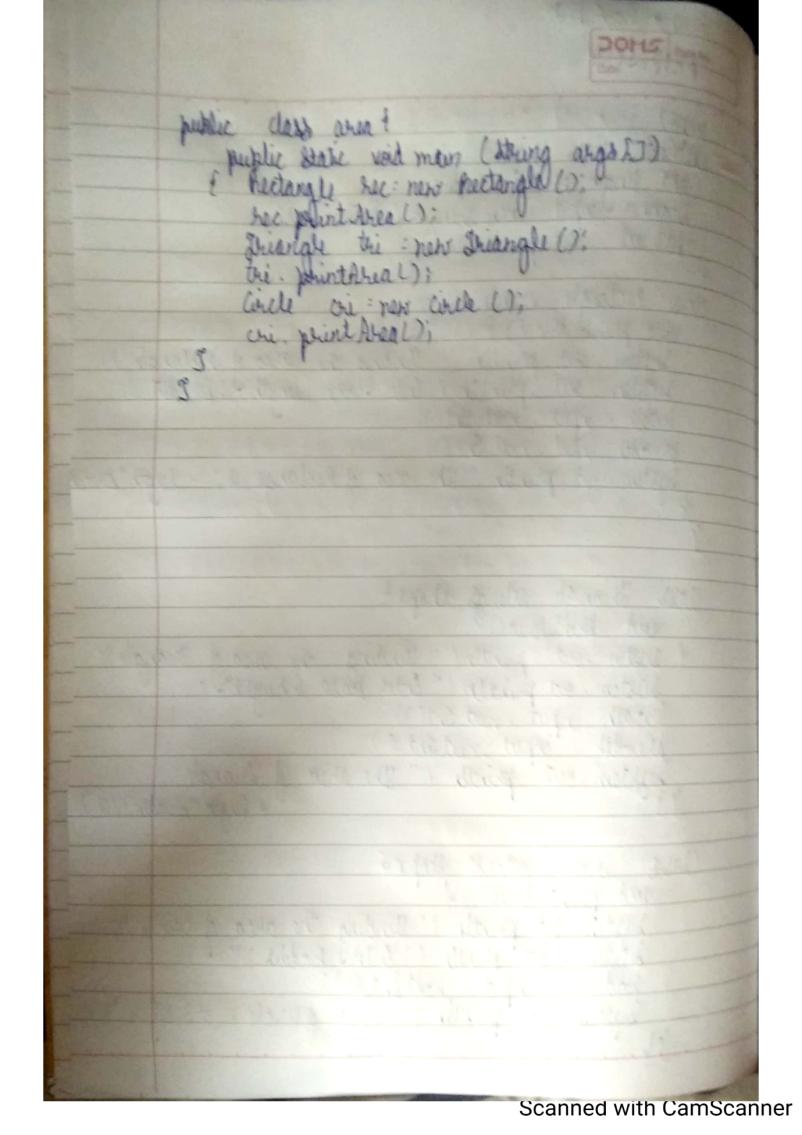
Develop a Java program to create an abstract class named Shape that contains two integers and

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.





**Source Code:**

**import java.util.\*;**

**abstract class Shape {**

**int length, breadth, radius;**

**Scanner input = new Scanner(System.in);**

**abstract void printArea();**

**}**

**class Rectangle extends Shape {**

**void printArea() {**

**System.out.println("\*\*\* Finding the Area of Rectangle \*\*\*");**

**System.out.print("Enter length and breadth: ");**

**length = input.nextInt();**

**breadth = input.nextInt();**

**System.out.println("The area of Rectangle is: " + length \* breadth);**

**}**

**}**

**+**

**class Triangle extends Shape {**

**void printArea() {**

**System.out.println("\n\*\*\* Finding the Area of Triangle \*\*\*");**

**System.out.print("Enter Base And Height: ");**

**length = input.nextInt();**

**breadth = input.nextInt();**

**System.out.println("The area of Triangle is: " + (length \* breadth) / 2);**

**}**

**}**

**class Cricle extends Shape {**

**void printArea() {**

**System.out.println("\n\*\*\* Finding the Area of Cricle \*\*\*");**

**System.out.print("Enter Radius: ");**

**radius = input.nextInt();**

**System.out.println("The area of Cricle is: " + 3.14f \* radius \* radius);**

**}**

**}**

**public class area {**

**public static void main(String[] args) {**

**Rectangle rec = new Rectangle();**

**rec.printArea();**

**Triangle tri = new Triangle();**

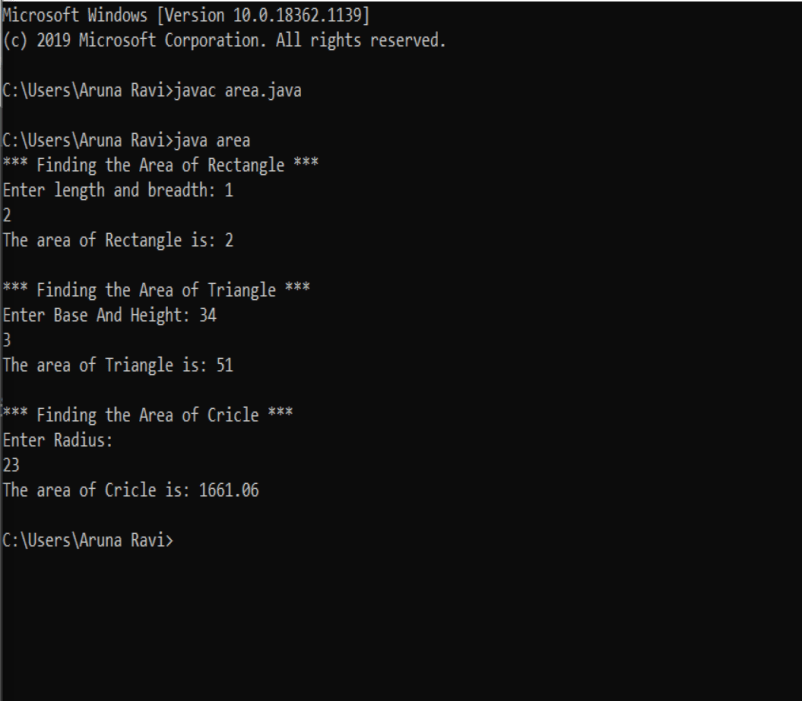
**tri.printArea();**

**Cricle cri = new Cricle();**

**cri.printArea();**

**}**

**}**



**Lab Exercises –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

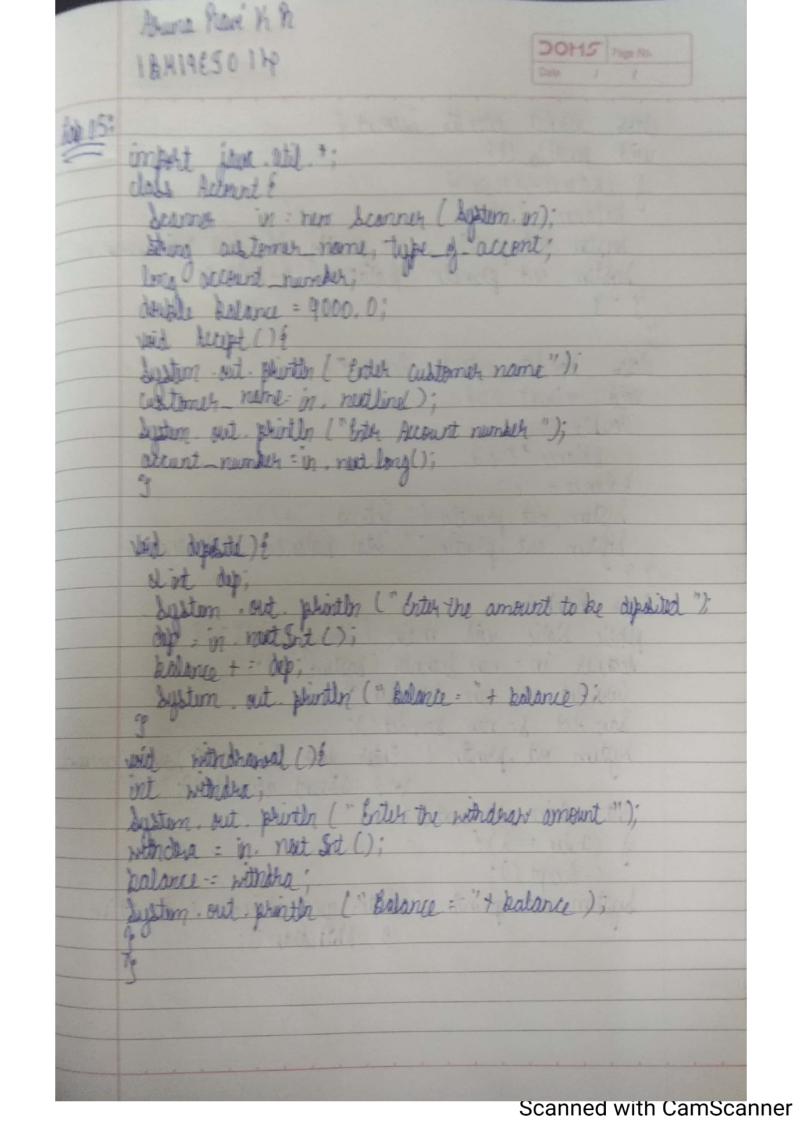
account. From this derive the classes Curr-acct and Sav-acct to make them more specific to

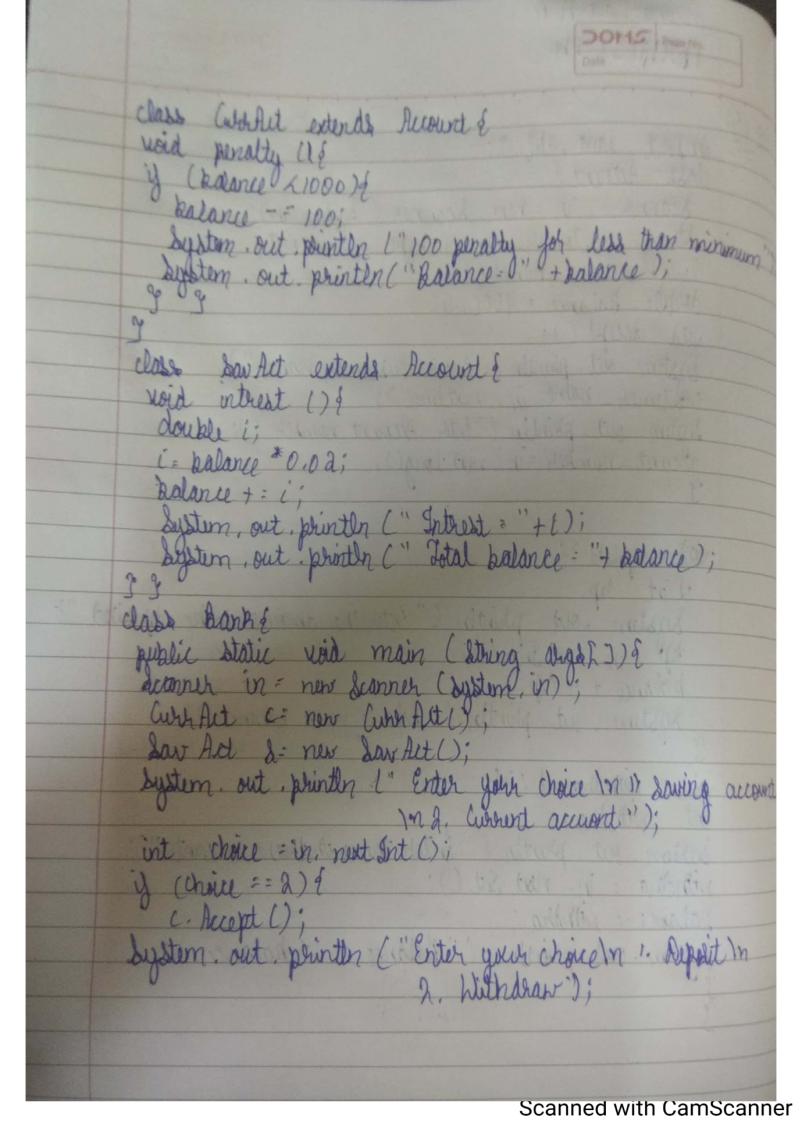
their requirements. Include the necessary methods in order to achieve the following tasks: •

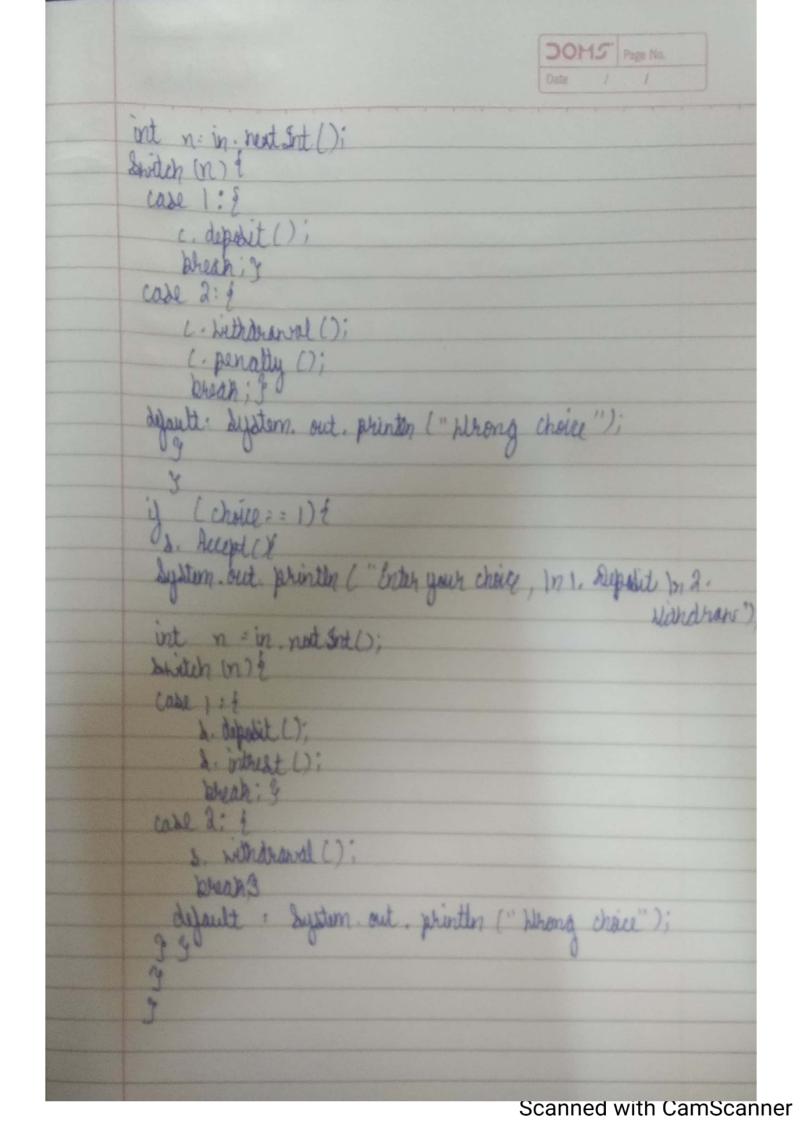
Accept deposit from customer and update the balance. • Display the balance. • Compute and

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

impose penalty if necessary and update the balance







**Source Code:**

**import java.util.\*;**

**class Account**

**{**

**Scanner in=new Scanner(System.in);**

**String customer\_name,type\_of\_account;**

**long account\_number;**

**double balance=9000;**

**void Accept(){**

**System.out.println("Enter customer name");**

**customer\_name=in.next();**

**System.out.println("Enter Account number");**

**account\_number=in.nextLong();**

**}**

**void deposit()**

**{**

**int dep;**

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

**dep=in.nextInt();**

**balance+=dep;**

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

**}**

**void withdrawal()**

**{**

**int witdra;**

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

**witdra=in.nextInt();**

**balance-=witdra;**

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

**}**

**}**

**class CurrAct extends Account{**

**void penalty(){**

**if(balance<2000){**

**balance-=100;**

**System.out.println("Penalty for maintaining less balance");**

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

**}**

**}**

**}**

**class SavAct extends Account{**

**void intrest(){**

**double i;**

**i=balance\*0.02;**

**balance+=i;**

**System.out.println("Intrest="+i);**

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

**}**

**}**

**class Bank{**

**public static void main (String args[]){**

**Scanner in=new Scanner(System.in);**

**CurrAct c=new CurrAct();**

**SavAct s=new SavAct();**

**System.out.println("Enter your choice\n1.Saving account\n2.current account");**

**int choice=in.nextInt();**

**if(choice==2){**

**c.Accept();**

**System.out.println("Enter your choice\n1. Deposite\n2.Withdrawl");**

**int n=in.nextInt();**

**switch(n){**

**case 1:{**

**c.deposit();**

**break;}**

**case 2:{**

**c.withdrawal();**

**c.penalty();**

**break;}**

**default : System.out.println("Wrong Option");**

**}**

**}**

**if(choice==1){**

**s.Accept();**

**System.out.println("Enter your choice\n1. Deposite\n2. Withdraw");**

**int n=in.nextInt();**

**switch (n){**

**case 1:{**

**s.deposit();**

**s.intrest();**

**break;}**

**case 2:{**

**s.withdrawal();**

**break;}**

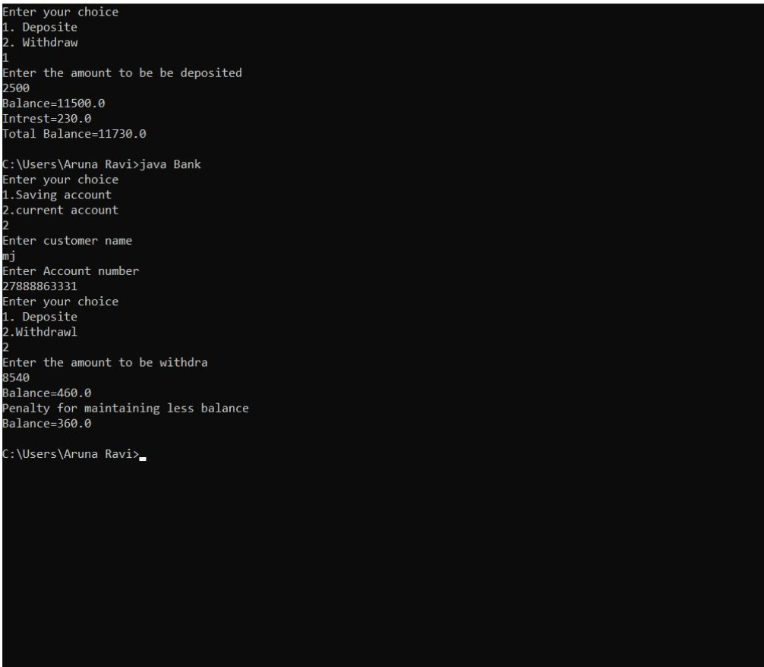
**default : System.out.println("Wrong option");**

**}**

**}**

**}**

**}**



**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

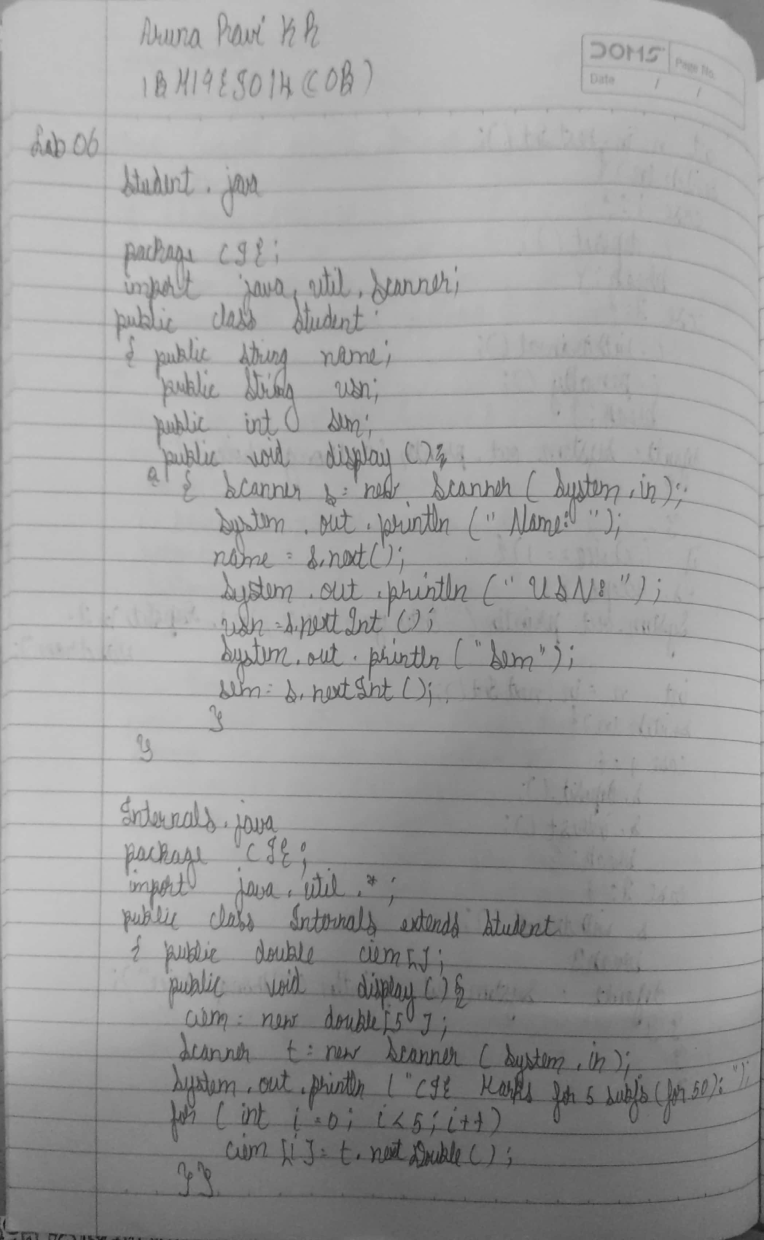
semester of the student. Create another package SEE which has the class

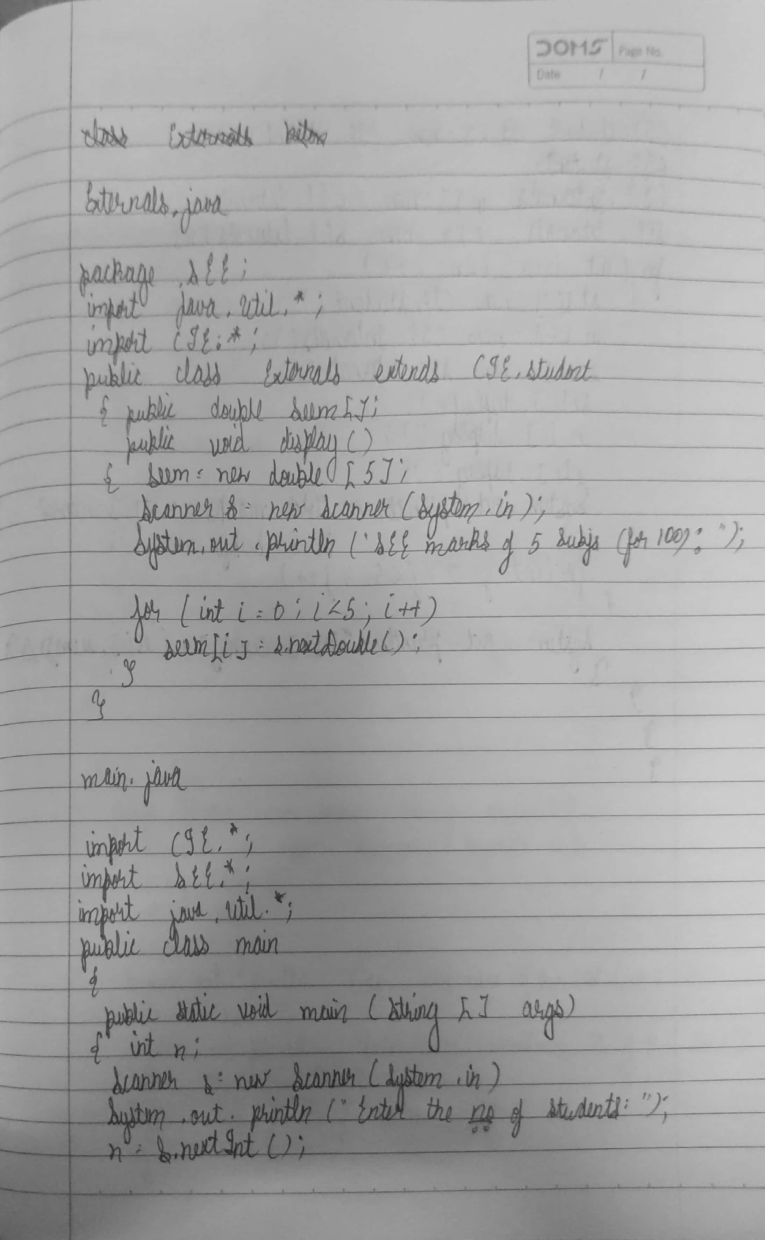
External which is a derived class of Student. This class has an array that

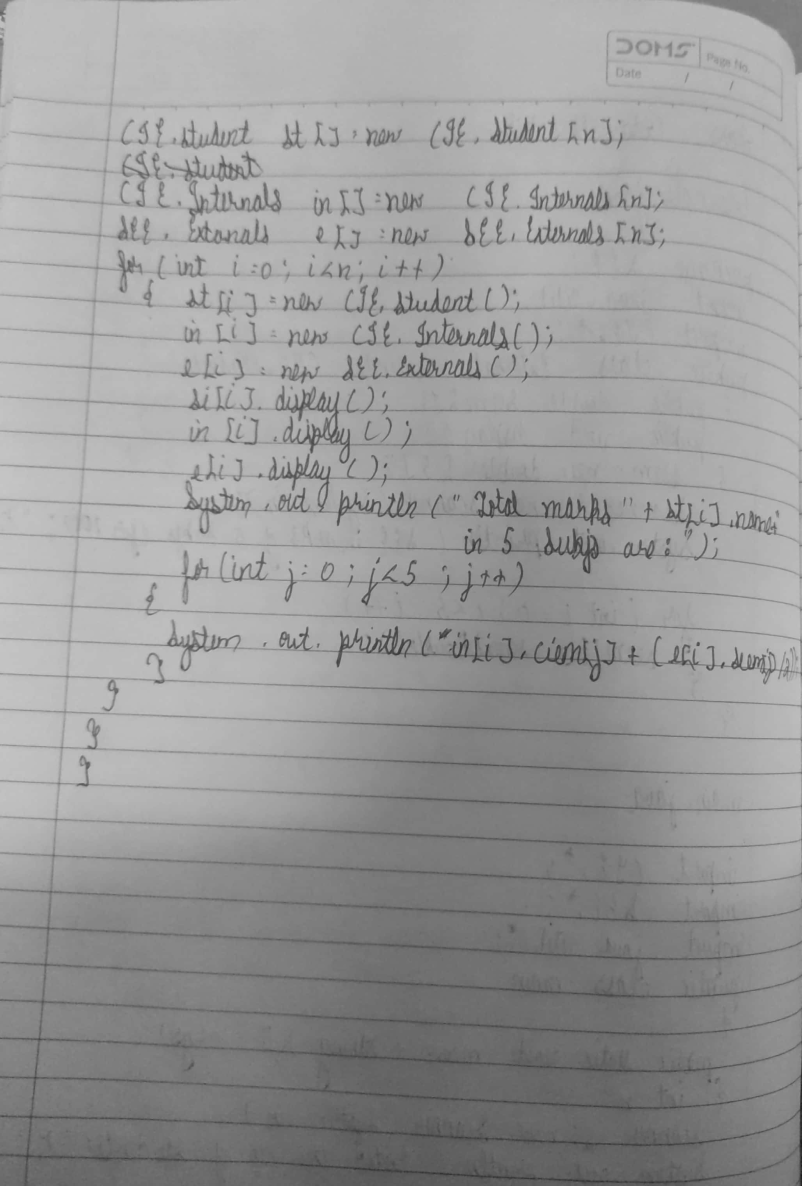
stores the SEE marks scored in five courses of the current semester of the

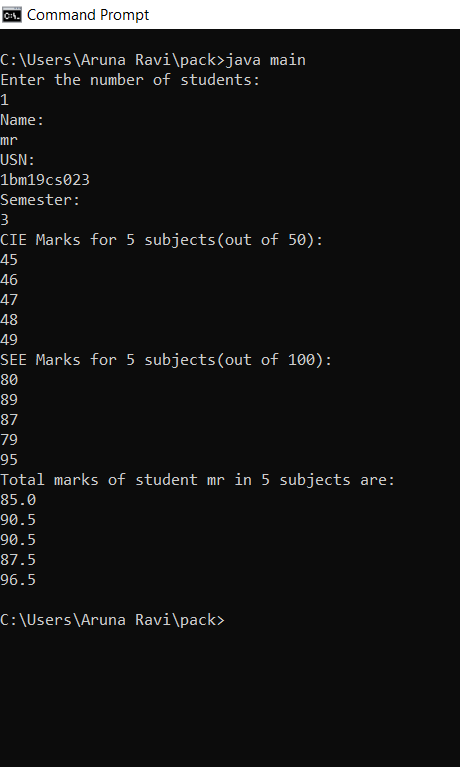
student. Import the two packages in a file that declares the final marks of n

students in all five courses.



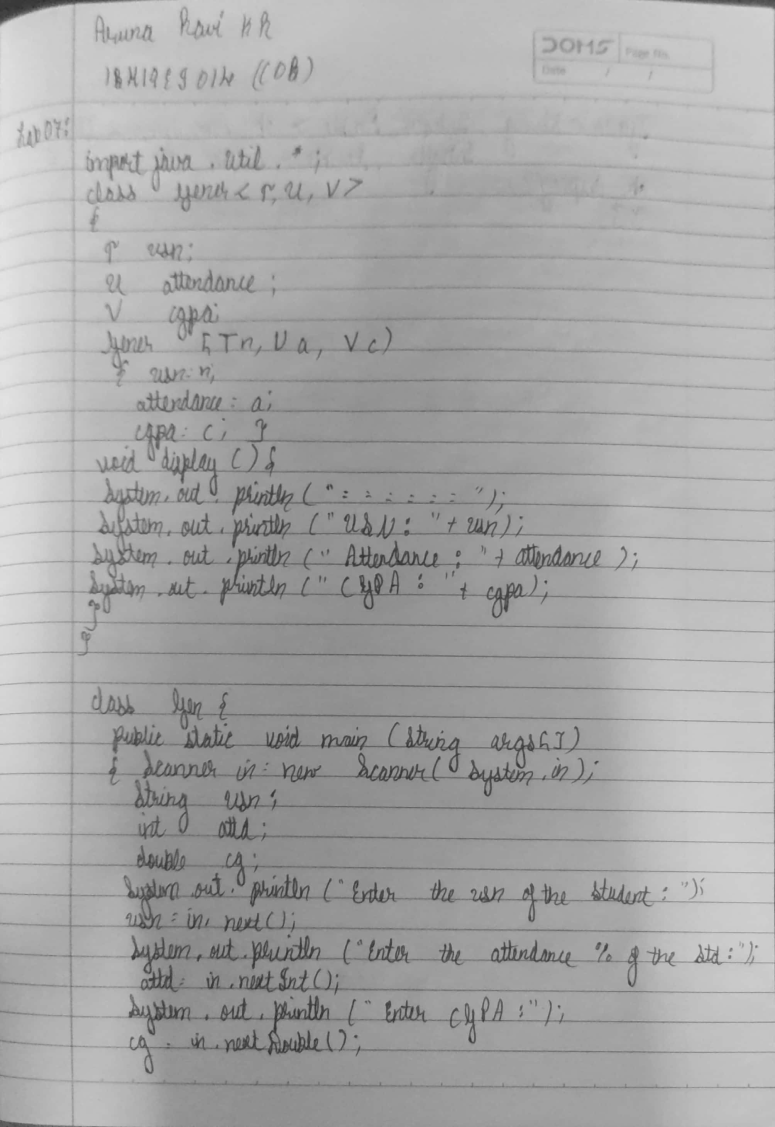


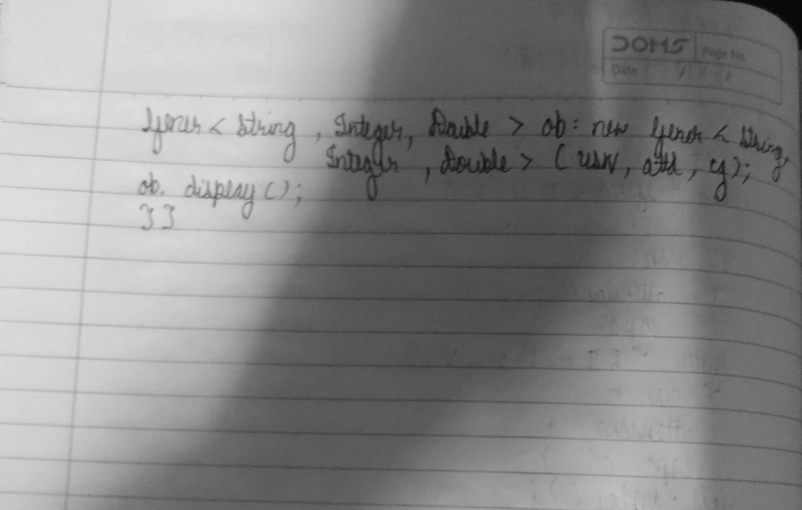




**Lab Programs-7**

Write a program to demonstrate generics with multiple object parameters.





**Source Code:**

**import java.util.\*;**

**class Gener<T,U,V>**

**{**

**T usn;**

**U attendance;**

**V cgpa;**

**Gener(T n,U a,V c)**

**{**

**usn = n;**

**attendance = a;**

**cgpa = c;**

**}**

**void display()**

**{**

**System.out.println("=============================");**

**System.out.println("USN of student: "+usn);**

**System.out.println("Attendance = "+attendance);**

**System.out.println("CGPA = "+cgpa);**

**}**

**}**

**class Lab07**

**{**

**public static void main(String args[])**

**{**

**Scanner in = new Scanner(System.in);**

**String USN;**

**int attd;**

**double cg;**

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

**USN = in.next();**

**System.out.println("Enter the attendance % of the student:");**

**attd = in.nextInt();**

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

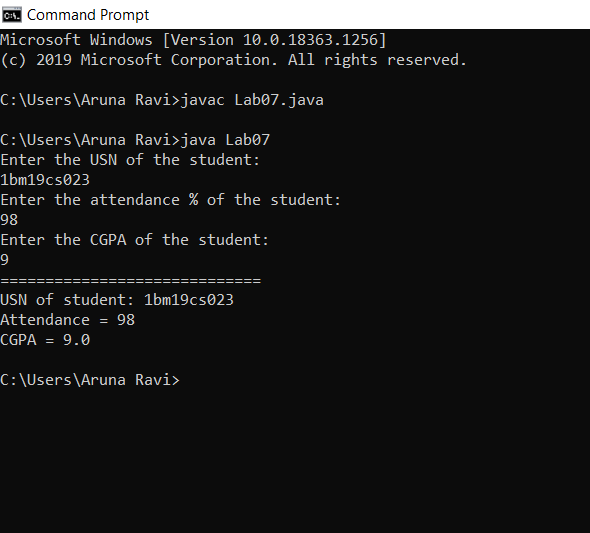
**cg = in.nextDouble();**

**Gener<String, Integer, Double> ob = new Gener<String, Integer, Double>(USN, attd, cg);**

**ob.display();**

**}**

**}**



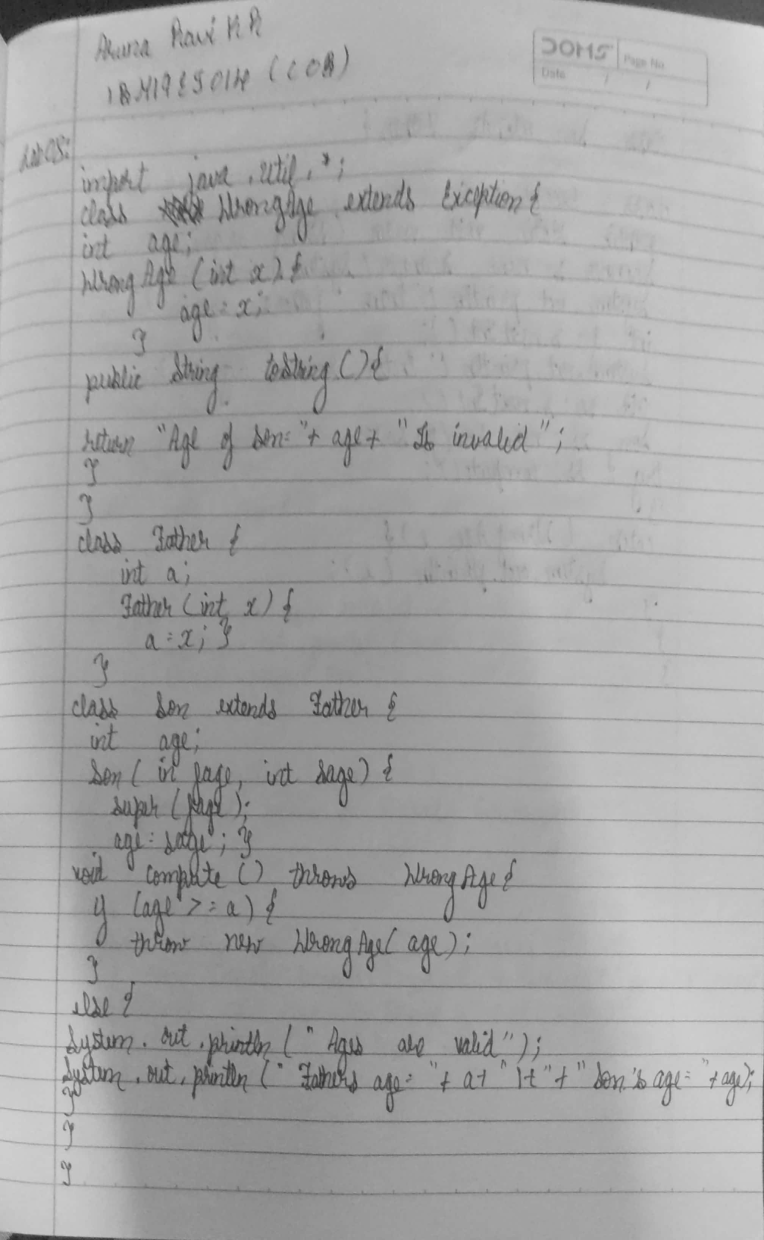
**Week 10: Lab Programs- 8:**

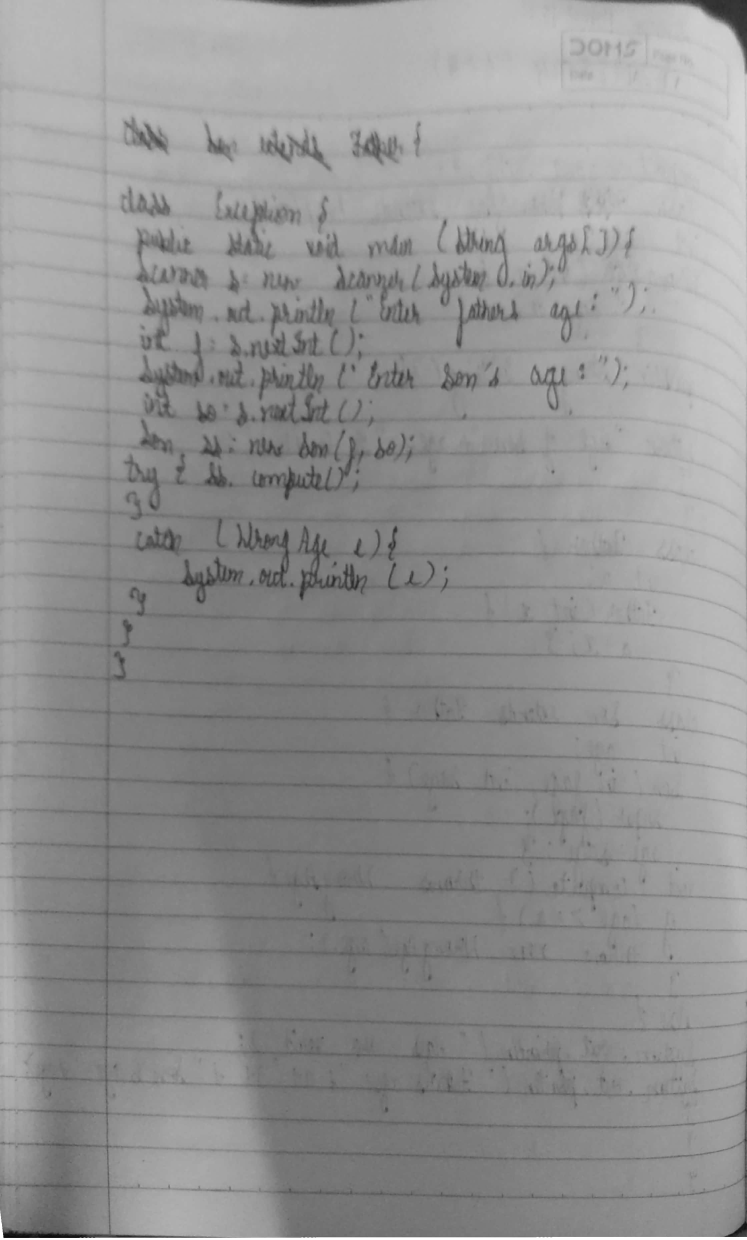
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class

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 Wrong Age( ) when the input

age=father’s age.





**Source Code:**

**import java.util.Scanner;**

**class WrongAge extends Exception {**

**int age;**

**WrongAge(int x) {**

**age = x;**

**}**

**public String toString() {**

**return "AGE OF SON=" + age + " IS INVALID";**

**}**

**}**

**class Father {**

**int a;**

**Father(int x) {**

**a = x;**

**}**

**}**

**class Son extends Father {**

**int age;**

**Son(int fage, int sage) {**

**super(fage);**

**age = sage;**

**}**

**void compute() throws WrongAge {**

**if (age >= a) {**

**throw new WrongAge(age);**

**} else {**

**System.out.println("THE AGES ARE VALID");**

**System.out.println("FATHER'S AGE=" + a + "\t" + "SON'S AGE=" + age);**

**}**

**}**

**}**

**class Exception {**

**public static void main(String args[]) {**

**Scanner s = new Scanner(System.in);**

**System.out.println("ENTER FATHER'S AGE:");**

**int f = s.nextInt();**

**System.out.println("ENTER SON'S AGE:");**

**int so = s.nextInt();**

**Son ss = new Son(f, so);**

**try {**

**ss.compute();**

**} catch (WrongAge e) {**

**System.out.println(e);**

**}**

**}**

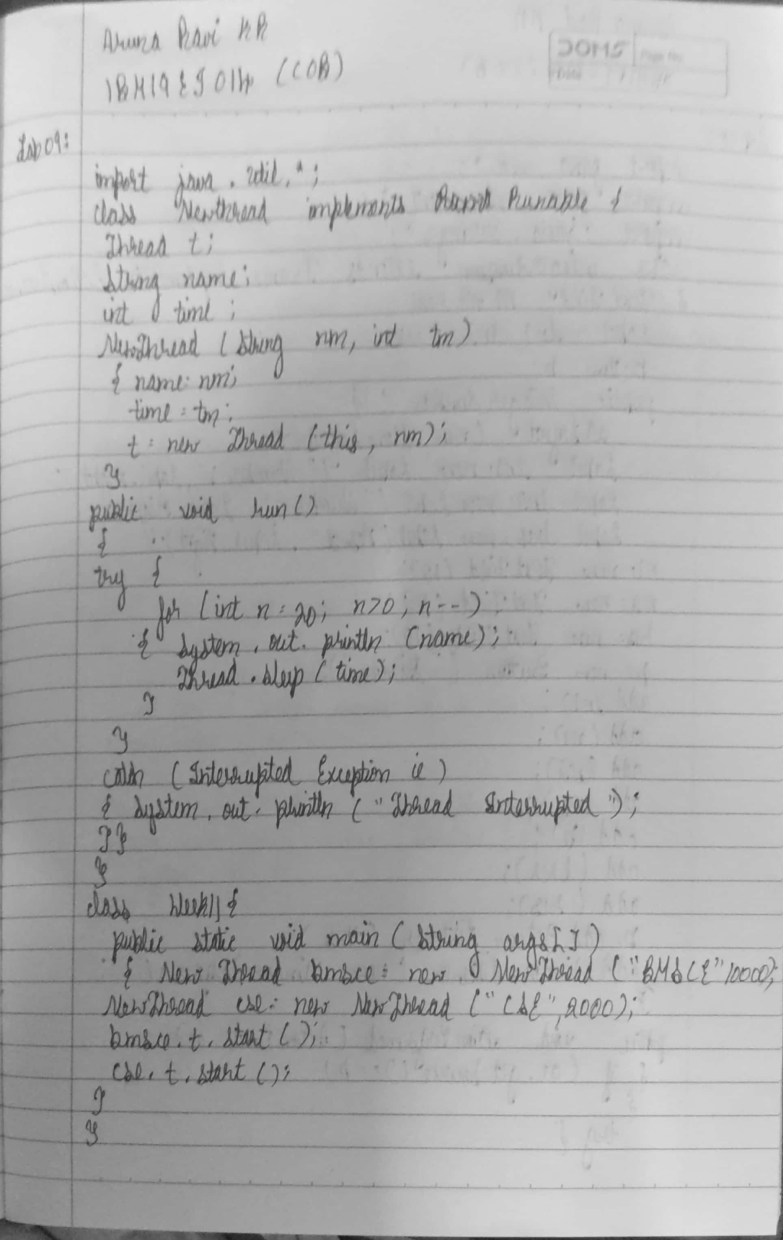
**}**



**LAB PROGRAM-09:**

**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.**



**Source Code:**

**class NewThread implements Runnable**

**{**

**Thread t;**

**String name;**

**int time;**

**NewThread(String nm,int tm)**

**{**

**name = nm;**

**time = tm;**

**t=new Thread(this, nm);**

**}**

**public void run()**

**{**

**try**

**{**

**for(int n=20;n>0;n--)**

**{**

**System.out.println(name);**

**Thread.sleep(time);**

**}**

**}**

**catch(InterruptedException ie)**

**{**

**System.out.println("Thread Interrupted");**

**}**

**}**

**}**

**class Lab09**

**{**

**public static void main(String ss[])**

**{**

**NewThread bmsce=new NewThread("BMS College of Engineering",10000);**

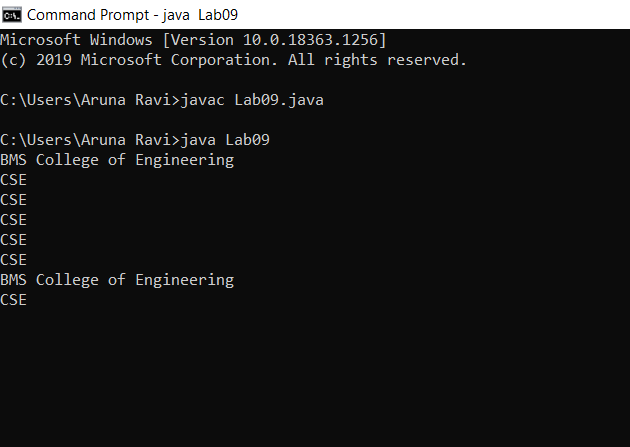
**NewThread cse=new NewThread("CSE",2000);**

**bmsce.t.start();**

**cse.t.start();**

**}**

**}**



**Lab Program-10:**

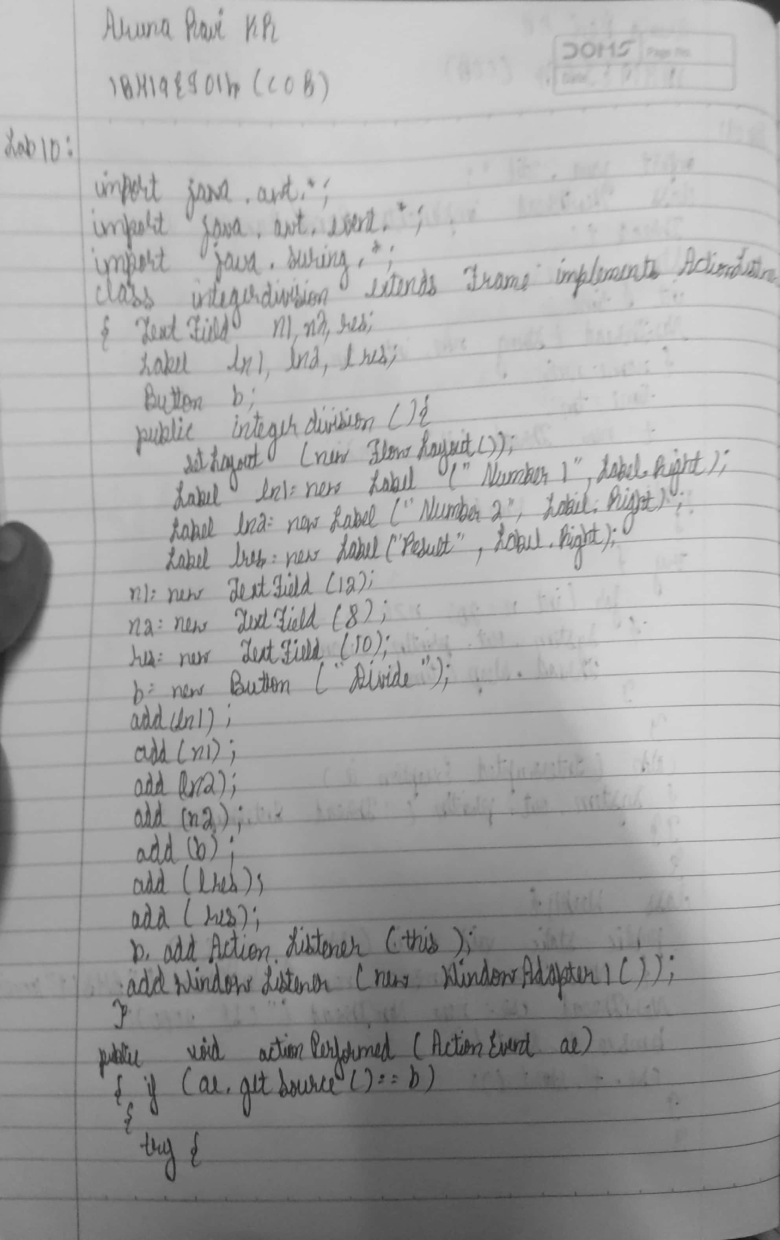
**Write a program that creates a user interface to perform integer divisions. The user enters two**

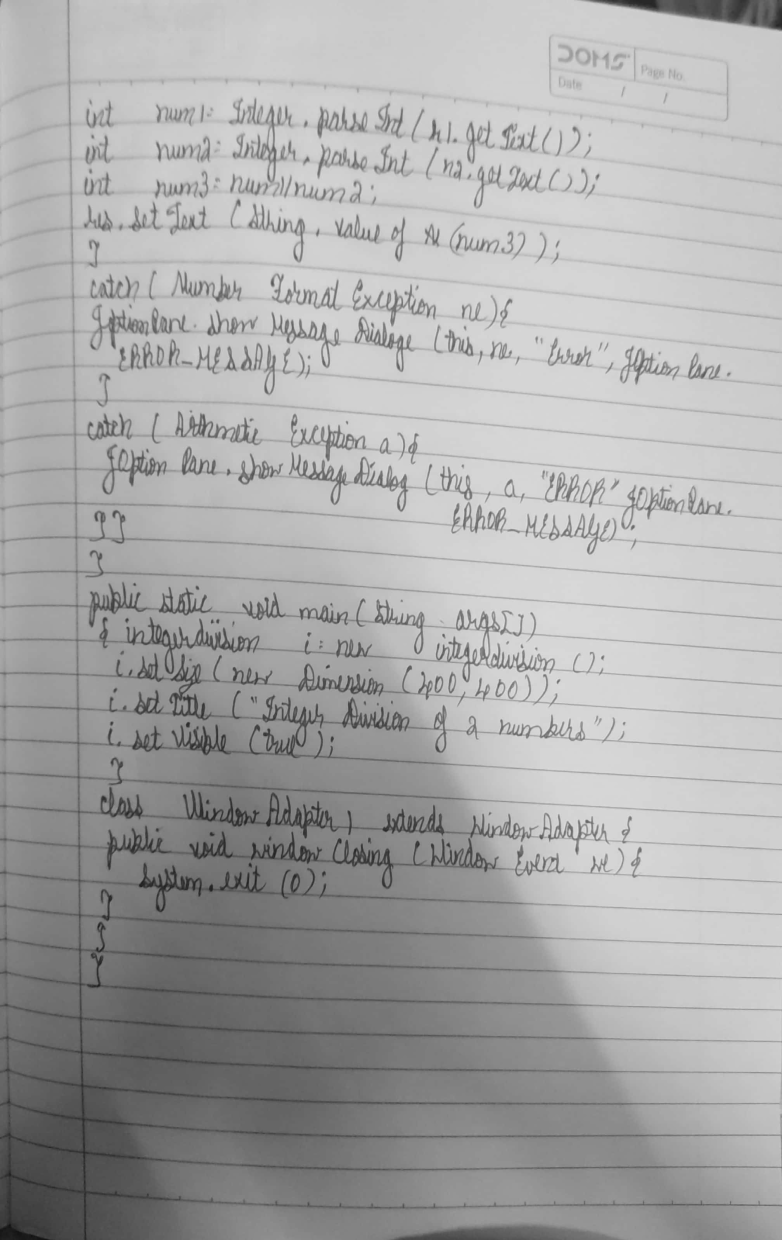
**numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the**

**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.**





**Source Code:**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import javax.swing.\*;**

**class integerdivision extends Frame implements ActionListener{**

**TextField n1,n2,res;**

**Label ln1,ln2,lres;**

**Button b;**

**public integerdivision(){**

**setLayout(new FlowLayout());**

**Label ln1=new Label("NUMBER 1",Label.RIGHT);**

**Label ln2=new Label("NUMBER 2",Label.RIGHT);**

**Label lres=new Label("RESULT",Label.RIGHT);**

**n1=new TextField(12);**

**n2=new TextField(8);**

**res=new TextField(10);**

**b=new Button("DIVIDE");**

**add(ln1);**

**add(n1);**

**add(ln2);**

**add(n2);**

**add(b);**

**add(lres);**

**add(res);**

**b.addActionListener(this);**

**addWindowListener(new WindowAdapter1());**

**}**

**public void actionPerformed(ActionEvent ae)**

**{**

**if(ae.getSource()==b)**

**{**

**try{**

**int num1=Integer.parseInt(n1.getText());**

**int num2=Integer.parseInt(n2.getText());**

**int num3=num1/num2;**

**res.setText(String.valueOf(num3));**

**}catch(NumberFormatException ne ){**

**JOptionPane.showMessageDialog(this,ne,"ERROR", JOptionPane.ERROR\_MESSAGE);**

**}**

**catch(ArithmeticException a){**

**JOptionPane.showMessageDialog(this,a,"ERROR", JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**}**

**public static void main(String args[])**

**{**

**integerdivision i=new integerdivision();**

**i.setSize(new Dimension(400,400));**

**i.setTitle("INTEGER DIVISION OF TWO NUMBERS");**

**i.setVisible(true);**

**}**

**class WindowAdapter1 extends WindowAdapter{**

**public void windowClosing(WindowEvent we)**

**{**

**System.exit(0);**

**}**

**}**

**}**

