**LAB REPORT**

NAME : NIRANJAN NAGARAJ SAVANUR USN : 1BM19CS104

SUBJECT : OOJ ACADEMIC YEAR : 2020-21

# 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.lang.Math; import java.util.Scanner;

public class Quadratic{

public static void main(String args[]){ Scanner sc = new Scanner(System.in);

System.out.println("-----Finding roots of a quadratic equation ");

System.out.println("Enter coefficients a, b, c of a QE in order w.r.t equation ax^2+bx+c=0");

double r, sqrt;

double a = sc.nextDouble(); double b = sc.nextDouble(); double c = sc.nextDouble();

double disc = (Math.pow(b, 2)) - 4 \* a \* c; if(disc < 0){

sqrt = (Math.sqrt(-disc))/ (2 \* a); r = -b /(2\*a);

System.out.println("Discriminant is negative. So no real roots are possible");

System.out.println("Imaginary roots are: " + r + " +i " + sqrt + " and " + r + " -i " + sqrt );

System.out.printf("or\n Imaginary roots are: %.2f +i %.4f and %.2f

-i %.4f", r, sqrt,r,sqrt);

}

else if(disc > 0){

sqrt = (Math.sqrt(disc)) / (2 \* a); r = -b / (2 \* a);

System.out.println("Real roots are: "+ (r+sqrt) + " and " + (r-sqrt));

System.out.printf("or\n Real roots are: %.4f and %.4f ", (r + sqrt), (r - sqrt));

}

else if(disc == 0){ r = -b / (2 \* a);

System.out.println("Roots are equal to: "+ r);

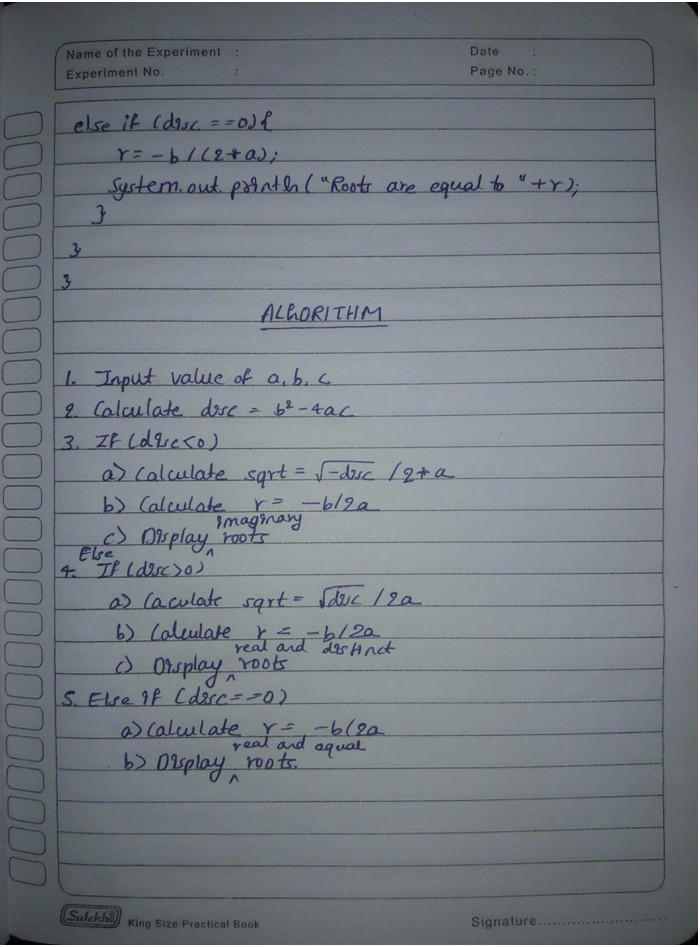
}

}

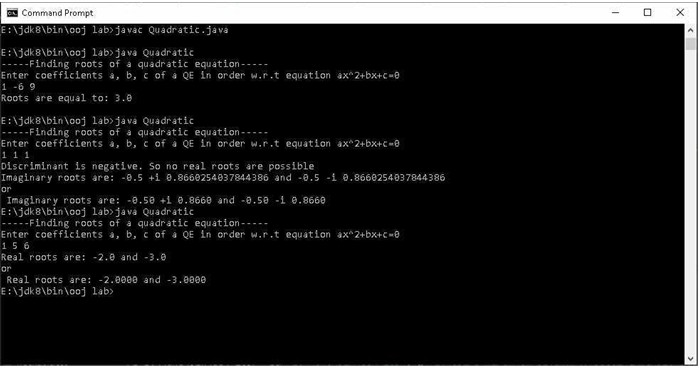
}

OBSERVATION:





OUTPUT:



# 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.Scanner; class Student{

private String usn; private String name;

private int[] credits = new int[20]; private int[] marks = new int[20]; private int n;

void getDetails(){

Scanner sc = new Scanner(System.in); System.out.println("Enter no of subjects:"); n = sc.nextInt();

System.out.println("Enter Student usn"); usn = sc.next();

System.out.println("Enter Student name");

name = sc.next();

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

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

credits[i] = sc.nextInt(); marks[i] = sc.nextInt();

}

}

void printDetails(){

System.out.println("Student details are as follows:"); System.out.println("Name: " + name + "\tusn: " + usn); for(int i = 0; i < n; i++){

System.out.println("Sub" + (i+1) + " Marks is: " + marks[i] + "\tCredit is: " + credits[i]);

}

}

void sgpaClac(){ double sgpa;

int[] gpcr = new int[n];

int credSum = 0, gp, sgpcr = 0; for(int i = 0; i < n; i++){ credSum += credits[i];

if (marks[i] >= 90){ gp = 10;

}

else if(marks[i] >= 80){ gp = 9;

}

else if(marks[i] >= 70){ gp = 8;

}

else if(marks[i] >= 60){ gp = 7;

}

else if(marks[i] >= 50){ gp = 5;

}

else if(marks[i] >= 40){ gp = 4;

}

else{ gp = 0;

}

gpcr[i] = gp \* credits[i]; sgpcr += gpcr[i];

}

sgpa = sgpcr / (credSum + 0.0); System.out.println("Student's sgpa is: " + sgpa);

}

}

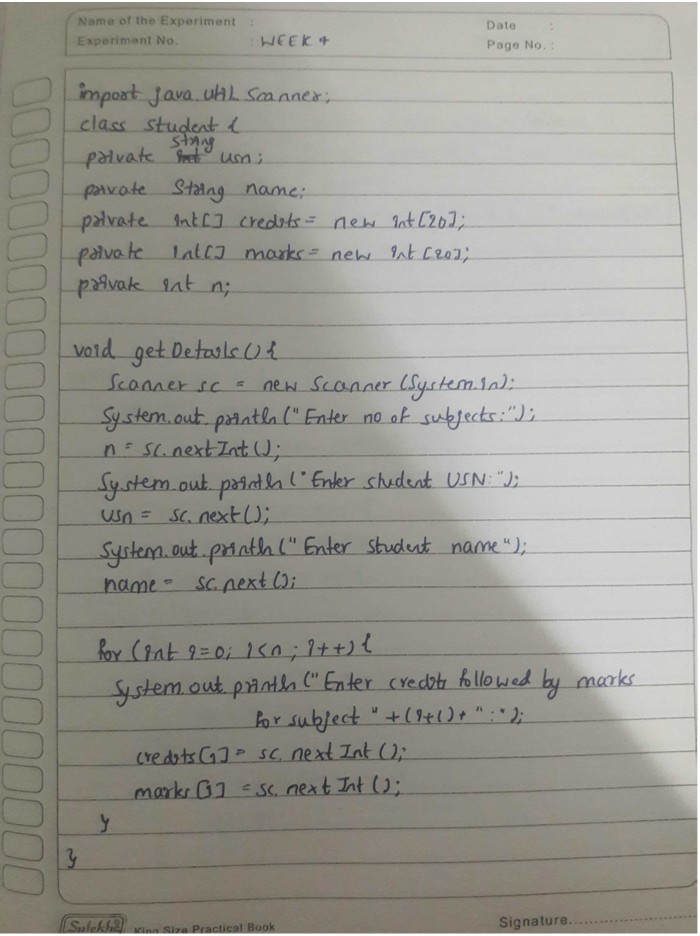
public class Main {

public static void main(String[] args){ Student s1 = new Student();

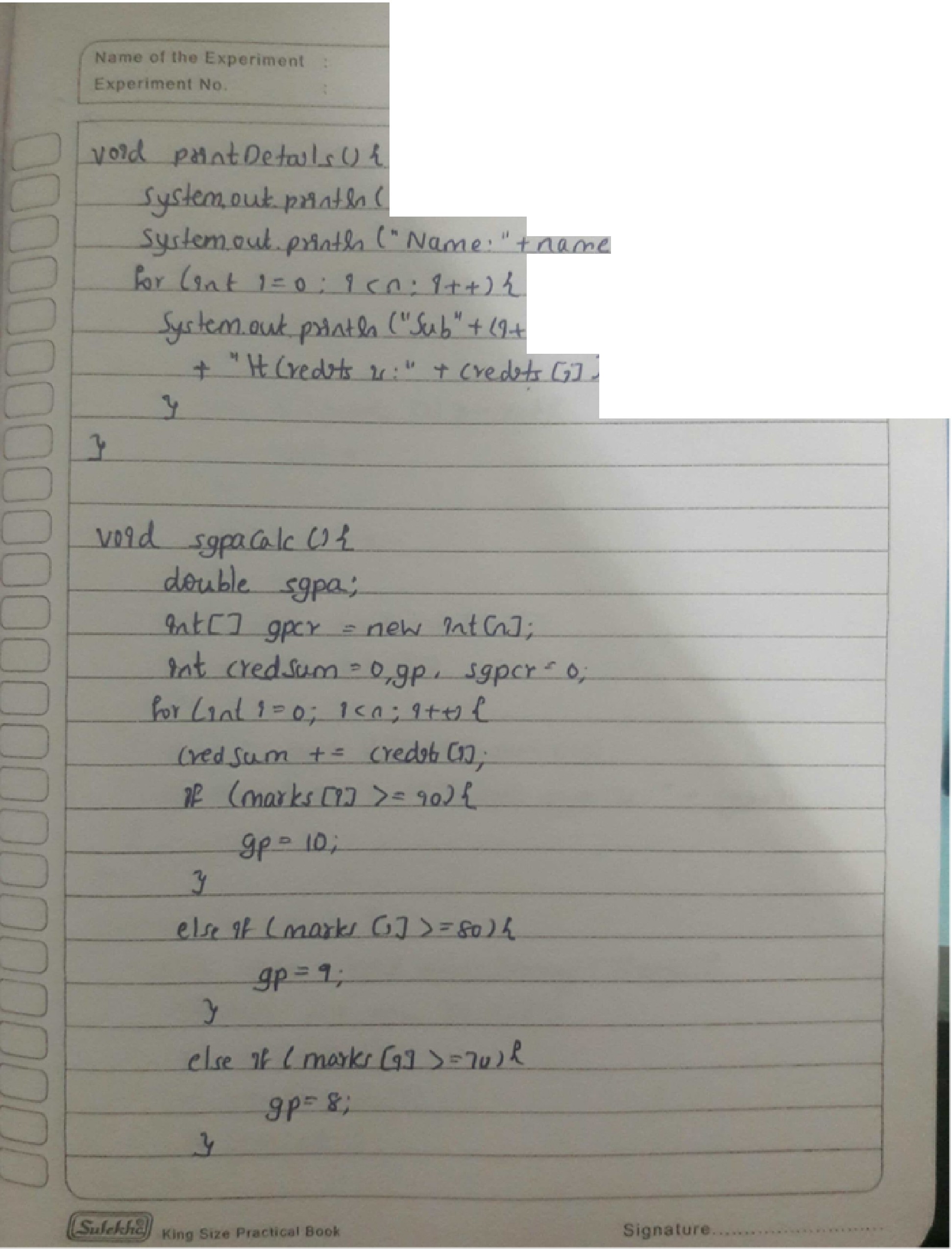
s1.getDetails(); s1.printDetails(); s1.sgpaClac();

}

} OBSERVATION:



DI



I

P, , No

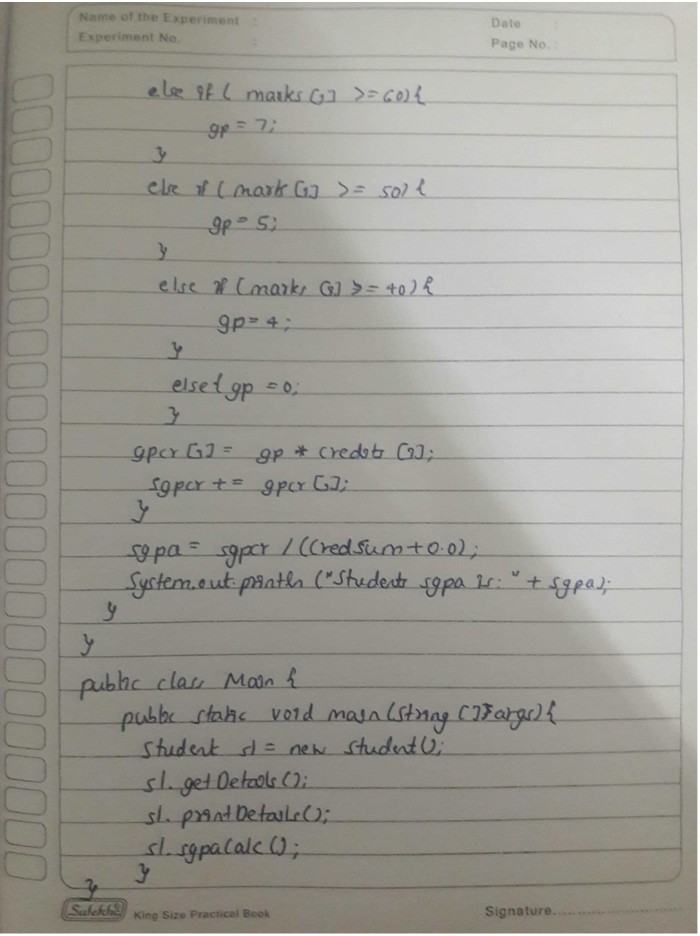
**----- ld..L 'JY,** cJ *w.,i.t 0 OJ f&tlr,w.s ),·* I

1

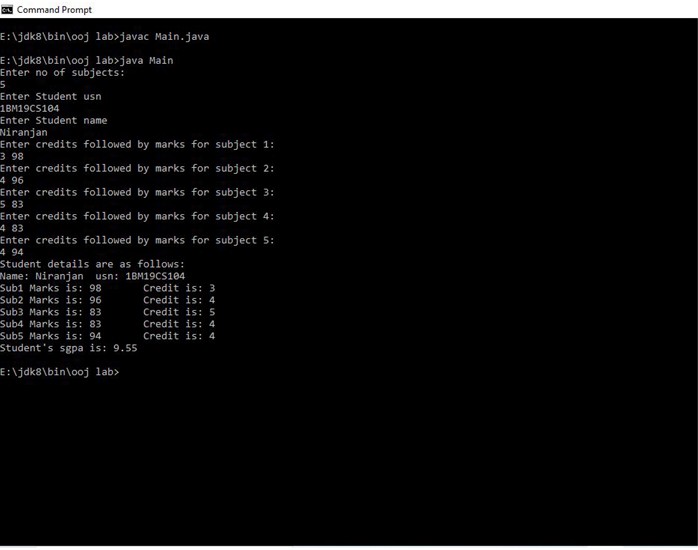
' ' } *VSn* • •• *VJ/\)*,*·*

IJ ,. Ma,l ***LJ*** *ma:,ksGJ*

*J*



OUTPUT:



# 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.Scanner; class Book{

private String author; private String name; private int num\_pages; private double price;

Scanner sc = new Scanner(System.in);

Book(){}

Book(String author, String name, int num\_pages, double price){ this.author = author;

this.name = name; this.num\_pages = num\_pages; this.price = price;

}

void getDetails(){ System.out.println("Enter author name"); author = sc.next(); System.out.println("Enter bookname"); name = sc.next(); System.out.println("Enter no of pages"); num\_pages = sc.nextInt(); System.out.println("Enter price");

price = sc.nextDouble();

}

public String toString(){

return ("AUTHOR :"+author+"\nBOOK NAME :"+name+"\nPAGES

:"+num\_pages+"\nPRICE :"+price);

}

}

class BookMain{

public static void main(String[] args){ Scanner sc = new Scanner(System.in); int n;

System.out.println("ENTER THE NUMBER OF BOOKS"); n = sc.nextInt();

System.out.println(" "); Book b[] = new Book[n];

b[0] = new Book("Tony", "Computers", 699, 2000.99 ); for(int i=1;i<n;i++){

System.out.println("ENTER DETAILS OF BOOK :"+(i+1)); b[i] = new Book();

b[i].getDetails();

System.out.println(" ");

}

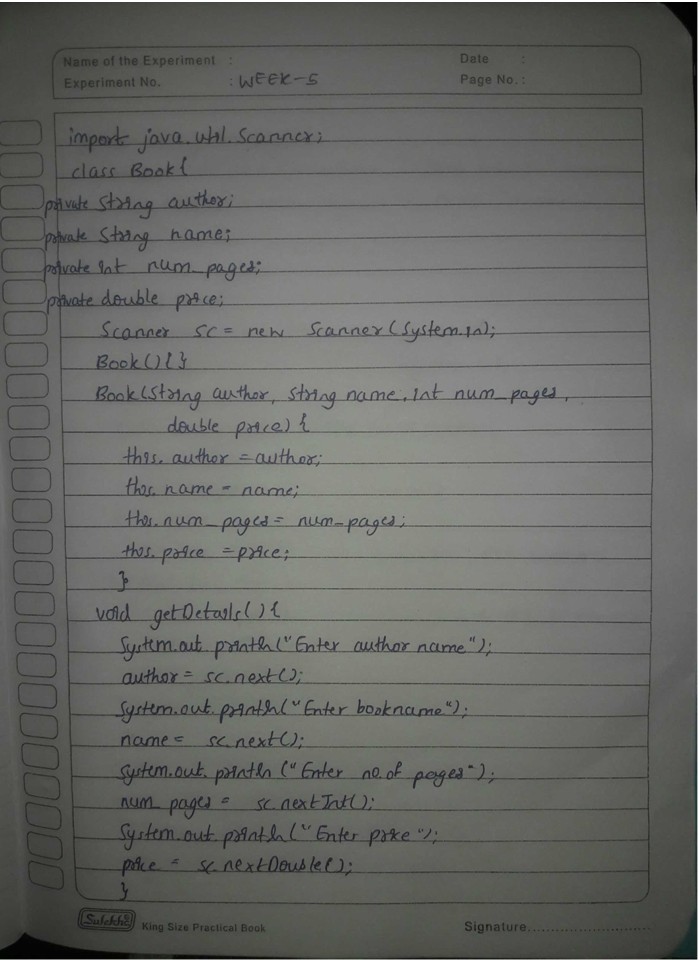
System.out.println("\n\nALL BOOK DETAILS THAT YOU ENTERED");

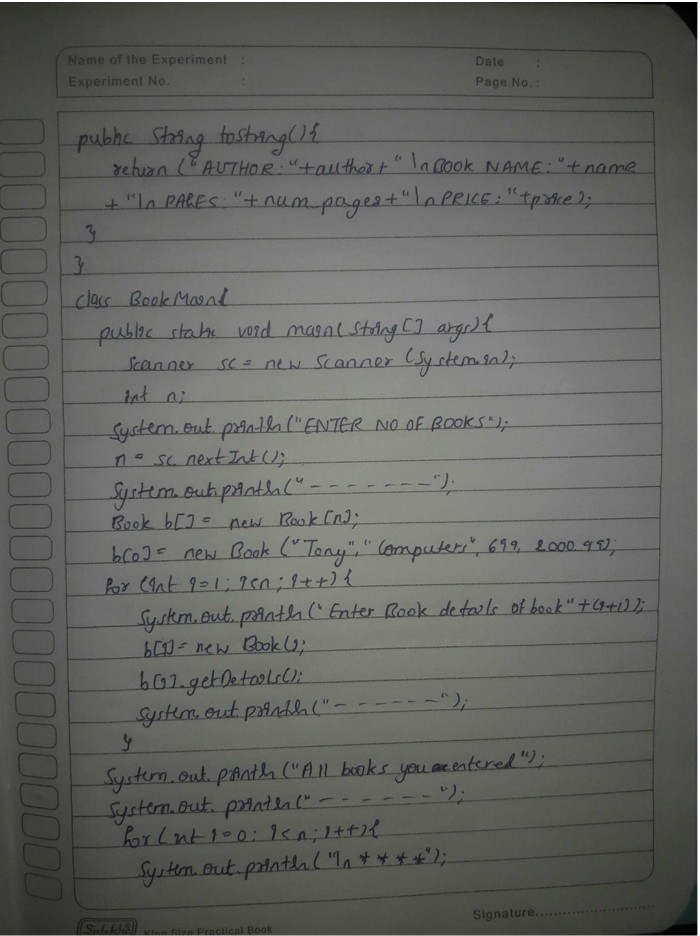
System.out.println(" "); for(int i=0;i<n;i++){ System.out.println("\n\*\*\*\*\*\*\*\*\*"); System.out.println("BOOOK :"+(i+1)); System.out.println(b[i]);

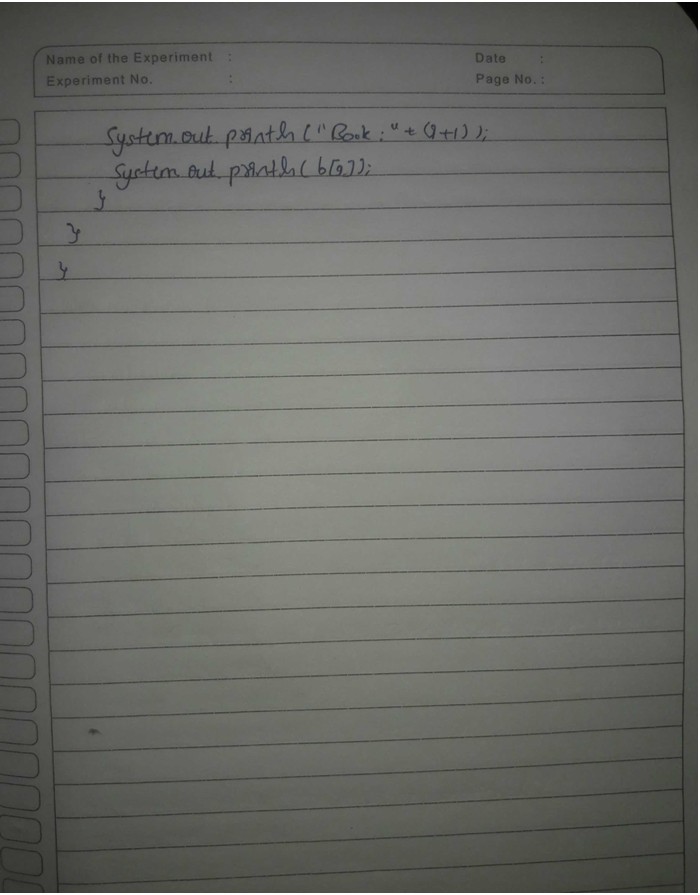
}

}

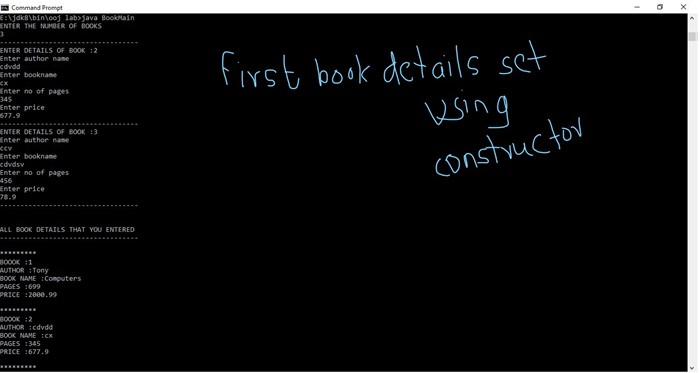
} OBSERVATION:







OUTPUT:



# 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.

SOURCE CODE:

import java.util.Scanner; import java.lang.Math;

abstract class Shape { int c, d;

Shape(int a, int b) { this.c = a;

this.d = b;

}

Shape(int a) { this.c = a;

}

abstract void printArea();

}

class Rectangle extends Shape { Rectangle(int a, int b) {

super(a, b);

}

void printArea() {

System.out.println("Area of rectangle is: " + c \* d);

}

}

class Triangle extends Shape { Triangle(int a, int b) {

super(a, b);

}

void printArea() {

System.out.println("Area of triangle is: " + c \* d / 2);

}

}

class Circle extends Shape { Circle(int a) {

super(a);

}

void printArea() {

System.out.println("Area of circle is: " + Math.PI \* Math.pow(c, 2));

}

}

public class shapeMain {

public static void main(String[] args){

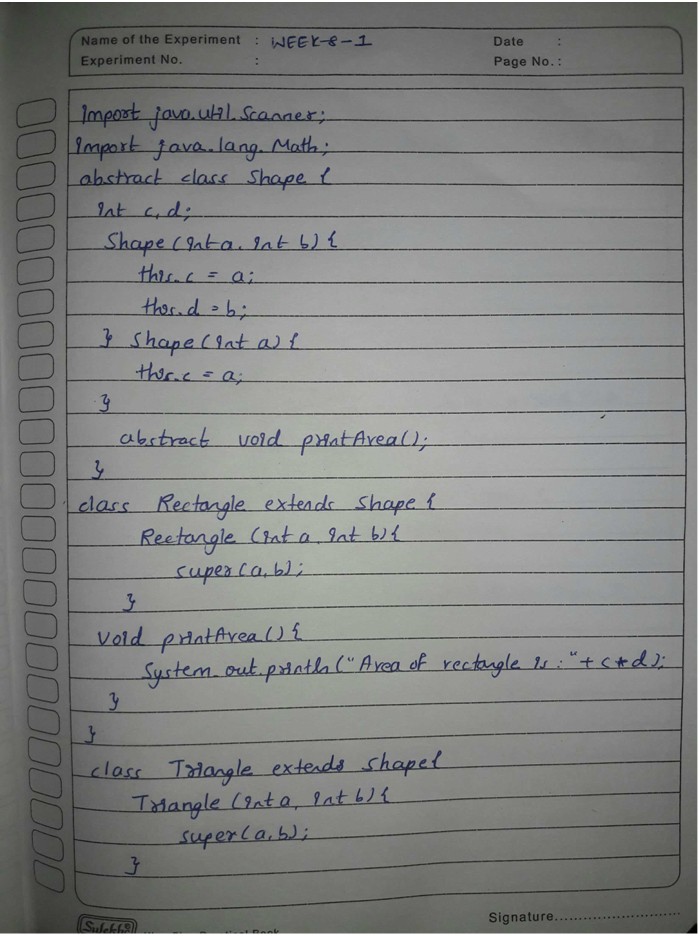
Scanner sc = new Scanner(System.in); Shape s;

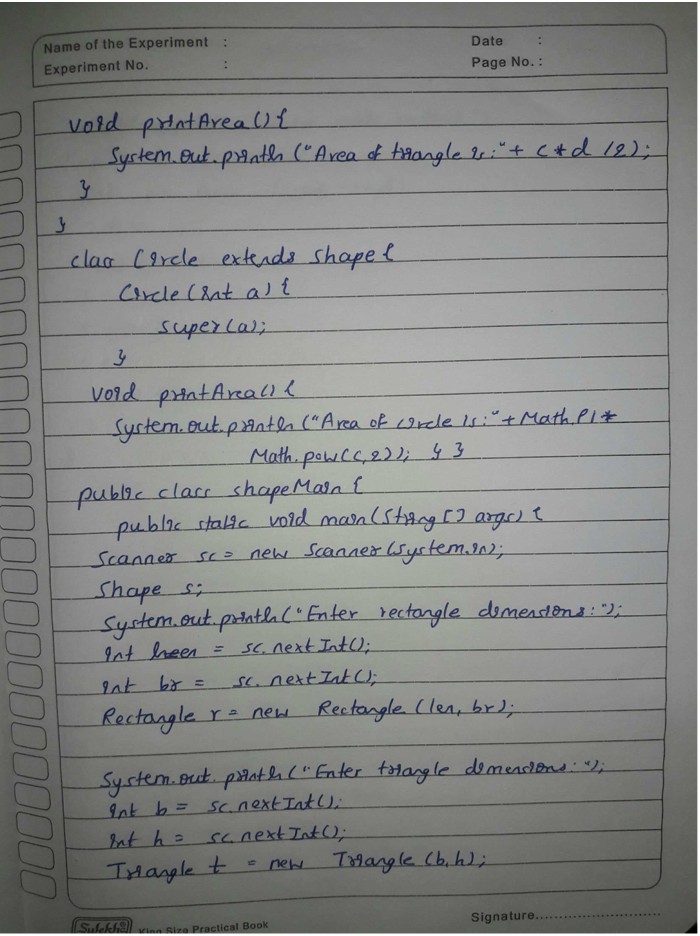
Rectangle r = new Rectangle(4, 10); Triangle t = new Triangle(6, 3); Circle c = new Circle(3);

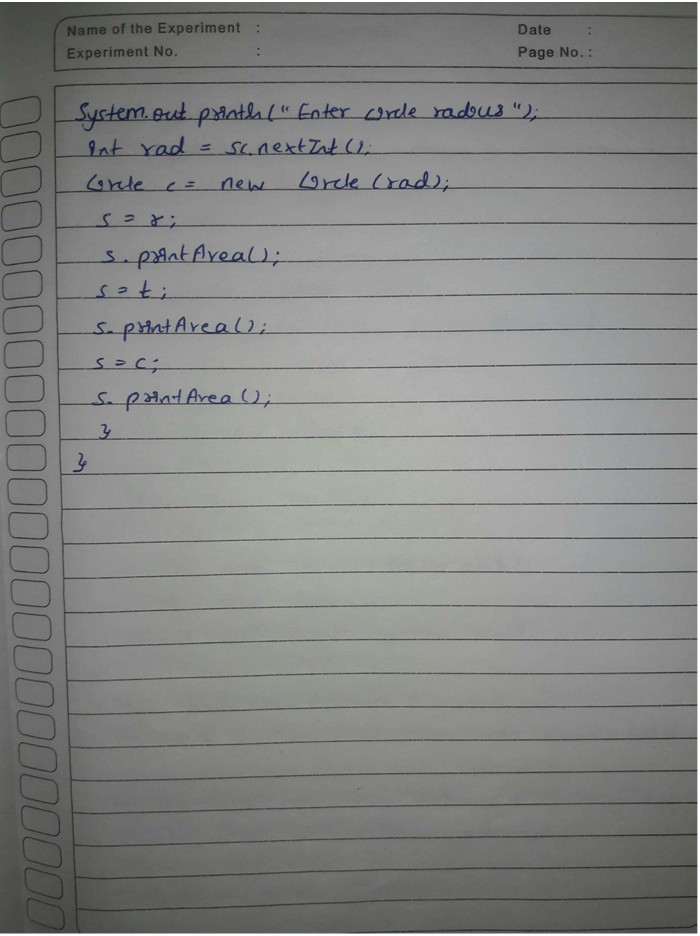
s = r; s.printArea(); s = t; s.printArea(); s = c; s.printArea();

}

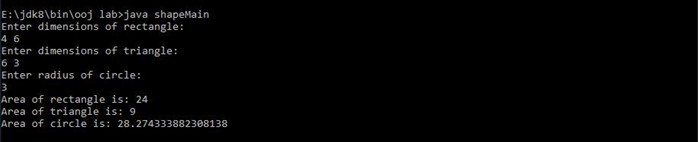
} OBSERVATION:







OUTPUT:



# 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:

·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.Scanner;

abstract class Account {

String cName, accType; long accNo;

double bal;

final double minBal = 1000.0;

Account(String cName, long accNo, double bal, String accType) { this.accNo = accNo;

this.cName = cName; this.bal = bal; this.accType = accType;

}

abstract void addBal(double amt);

abstract void dispBal();

abstract void withBal(double amt);

}

class Curr\_acct extends Account { Curr\_acct(String cName, long accNo, double bal) { super(cName, accNo, bal, "Current");

System.out.println("name: " + cName + "\taccno: " + accNo + "\tbal: " + bal + "\ttype: " + accType);

}

void addBal(double amt) { this.bal += amt;

}

void dispBal() {

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

}

void withBal(double amt) {

if (this.bal == 0 || amt > this.bal) { System.out.println("withdrawal not possible");

}else{

this.bal -= amt; checkBal();

}

}

void checkBal() {

if (this.bal < minBal) { this.bal -= this.bal \* 0.02;

}

}

}

class Sav\_acct extends Account { Sav\_acct(String cName, long accNo, double bal) { super(cName, accNo, bal, "Savings");

System.out.println("name: " + cName + "\taccno: " + accNo + "\tbal: " + bal + "\ttype: " + accType);

}

void addBal(double amt) {

this.bal += amt; addIntr();

}

void addIntr() {

this.bal += this.bal \* 0.07;

}

void dispBal() {

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

}

void withBal(double amt) {

if (this.bal == 0 || amt > this.bal) { System.out.println("withdrawal not possible");

}else{

this.bal -= amt;

}

}

}

class AccountMain {

public static void main(String[] args) { Scanner sc = new Scanner(System.in); Double amt;

int flag = 0;

while (flag == 0) {

System.out.println("1:Current acc.\n2:Savings acc.\ndefault:exit"); int ch = sc.nextInt();

String nam; long acno; double balan; switch (ch) { case 1:

System.out.println("Enter name, acc no, initial balance in order:"); nam = sc.next();

acno = sc.nextLong(); balan = sc.nextDouble();

Curr\_acct c = new Curr\_acct(nam, acno, balan); System.out.println("\nCurrent\_acct\n");

int flag1 = 0;

while (flag1 == 0) {

System.out.println("1:Addamount\n2:displayBalance\n3:withdraw\ ndefault:exit");

int ch1 = sc.nextInt(); switch (ch1) {

case 1:

System.out.println("enter amt to be added:"); amt = sc.nextDouble();

c.addBal(amt); break;

case 2: c.dispBal(); break;

case 3:

System.out.println("enter amt to be withdrawn:"); amt = sc.nextDouble();

c.withBal(amt); break;

default:

flag1 = 1;

}

}

break; case 2:

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

System.out.println("Enter name, acc no, initial balance in order:"); nam = sc.next();

acno = sc.nextLong(); balan = sc.nextDouble();

Sav\_acct s = new Sav\_acct(nam, acno, balan);

int flag2 = 0;

while (flag2 == 0) {

System.out.println("1:AddBal\n2:displayBal\n3:withdraw\ndefault: exit");

int ch2 = sc.nextInt(); switch (ch2) {

case 1:

System.out.println("enter amt to be added:"); amt = sc.nextDouble();

s.addBal(amt); break;

case 2:

s.dispBal(); break;

case 3:

System.out.println("enter amt to be withdrawn:"); amt = sc.nextDouble();

s.withBal(amt);

break;

default: flag2 = 1;

}

}

break; default:

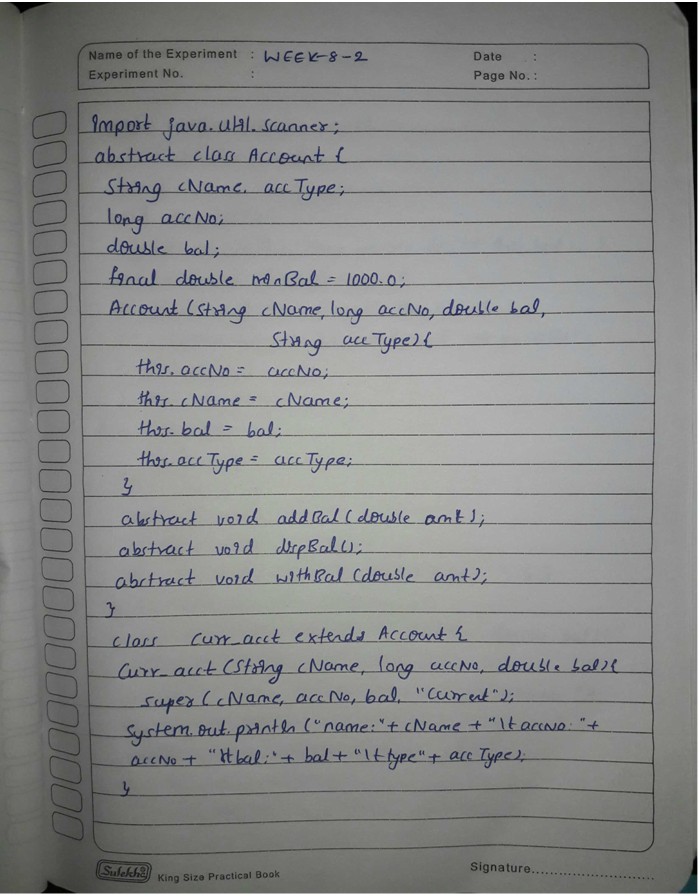
flag = 1;

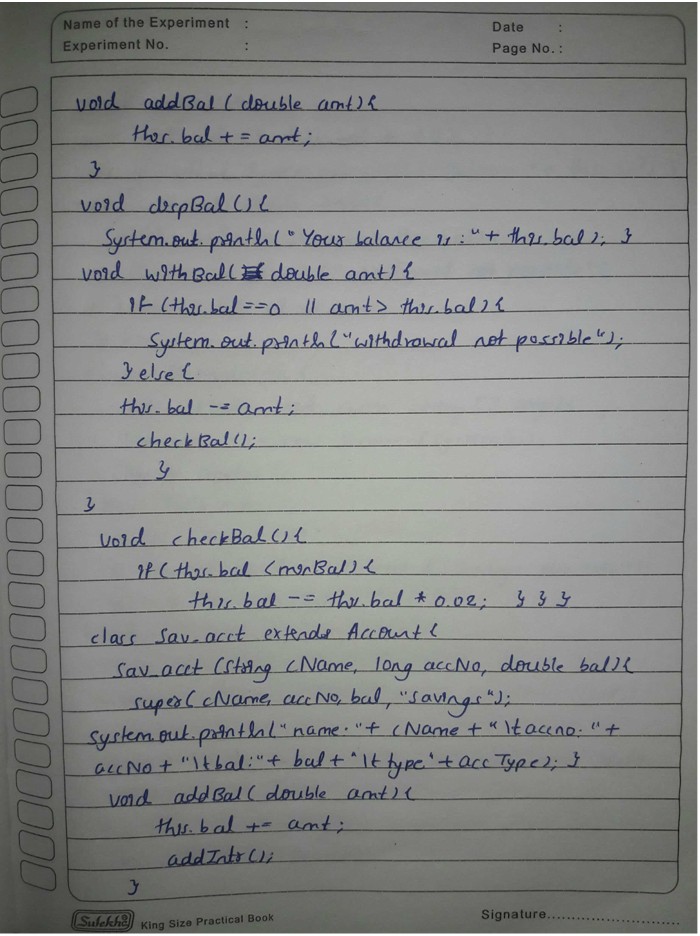
}

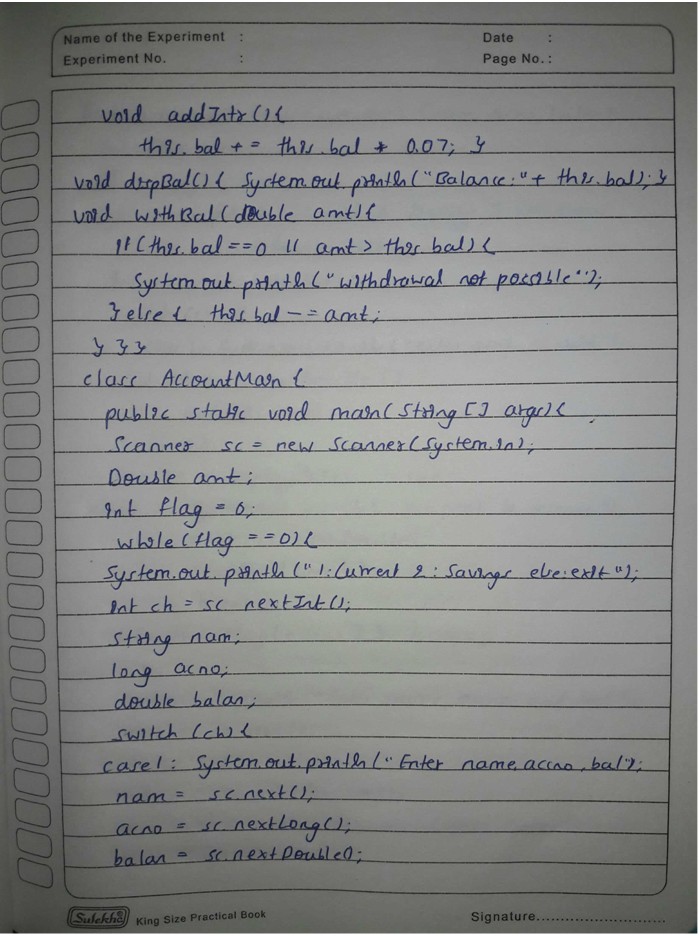
}

}

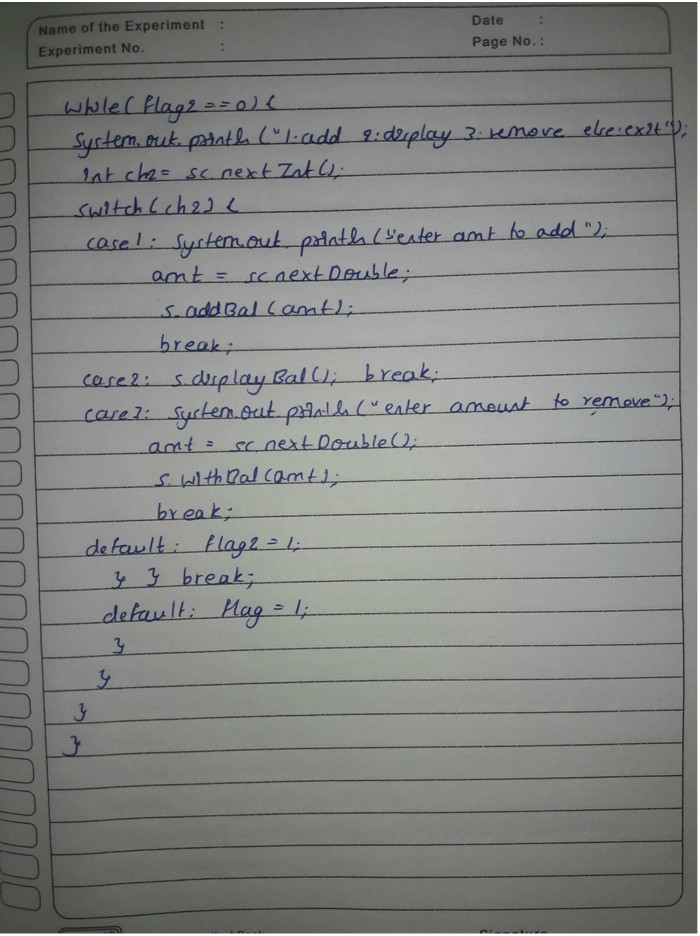
} OBSERVATION:



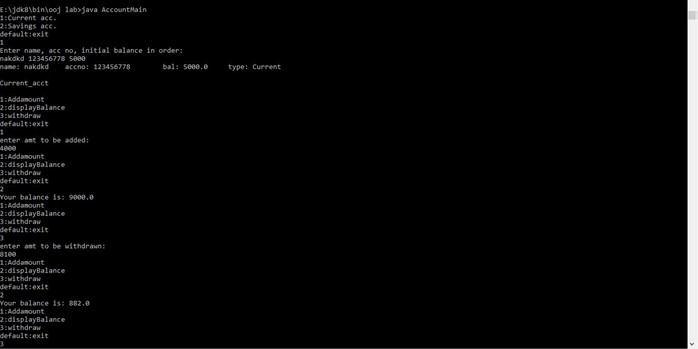


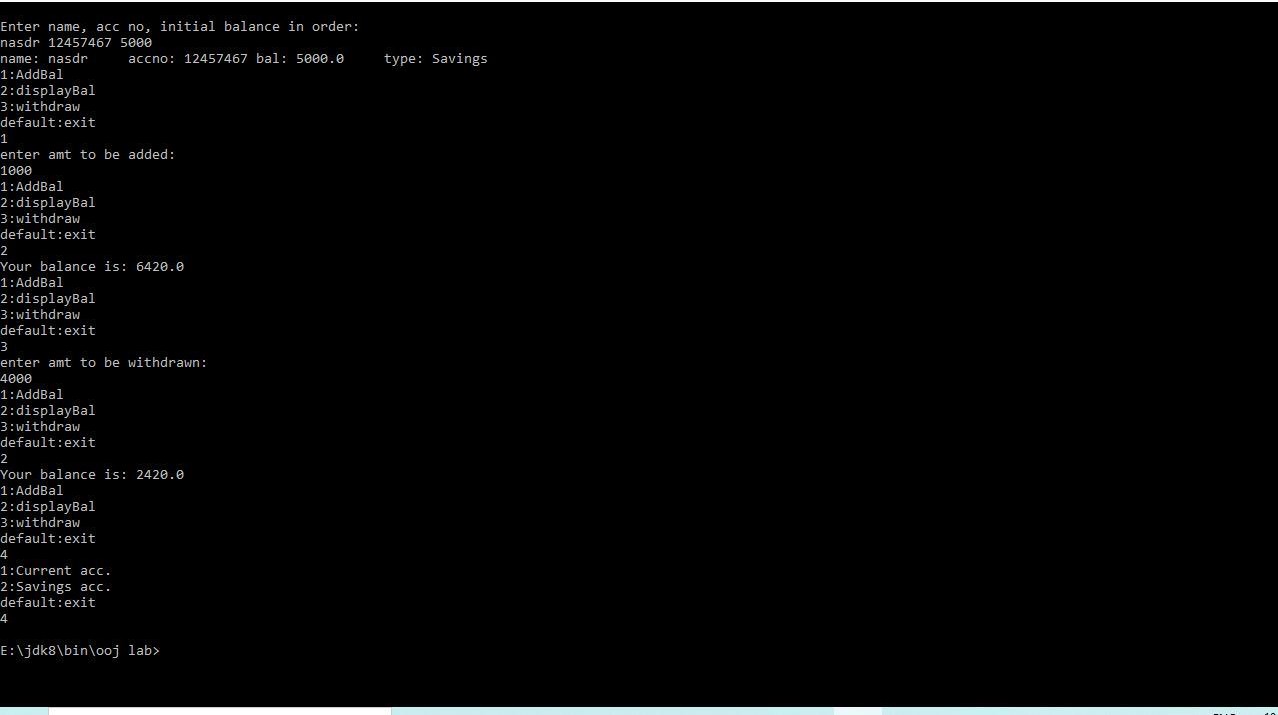






OUTPUT:





# 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.

SOURCE CODE:

import CIE.\*; import SEE.\*;

import java.util.Scanner;

class finalmarks {

public static void main(String args[]) { Scanner s = new Scanner(System.in);

System.out.println("ENTER THE NUMBER OF STUDENTS"); int n = s.nextInt();

SEE.Externals ob1[] = new SEE.Externals[n]; CIE.Internals ob[] = new CIE.Internals[n];

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

System.out.println("\nENTER THE USN,NAME AND SEMESTER OF STUDENT" + (i + 1));

String u = s.next(); String nm = s.next(); int se = s.nextInt();

ob1[i] = new SEE.Externals(u, nm, se);

ob[i] = new CIE.Internals(); ob[i].get();

ob1[i].get();

}

for (int i = 0; i < n; i++) { System.out.println("\nSTUDENT\_" + (i + 1));

System.out.println("name = " + ob1[i].name +"\tusn = " + ob1[i].usn + "\tsem = " + ob1[i].sem);

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

System.out.println("SUBJECT\_" + (j + 1) + "\_MARKS = " + (ob[i].cie[j] + (ob1[i].see[j])/2));

}

}

}

}

PACKAGE SEE:-

package SEE;

import java.util.Scanner;

public class Externals extends CIE.Student { public Externals(String usn, String name, int sem) { super(usn, name, sem);

}

Scanner s = new Scanner(System.in); public int[] see = new int[5];

public void get() {

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

System.out.println("ENTER THE SEE MARK IN SUBJECT" + (i + 1));

see[i] = s.nextInt();

}

}

}

PACKAGE CIE:-

package CIE;

import java.util.Scanner;

public class Internals extends CIE.Student { Scanner s = new Scanner(System.in); public int[] cie = new int[5];

public void get() {

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

System.out.println("ENTER THE CIE MARK IN SUBJECT" + (i + 1));

cie[i] = s.nextInt();

}

}

}

package CIE;

public class Student { public String usn; public String name; public int sem;

public Student() {

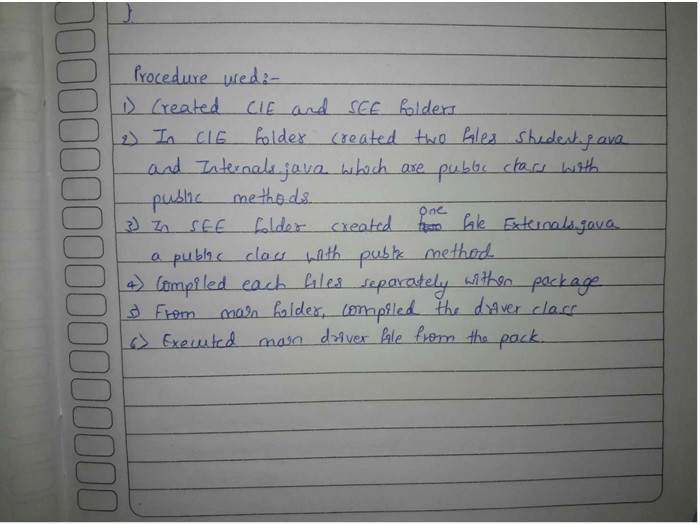
}

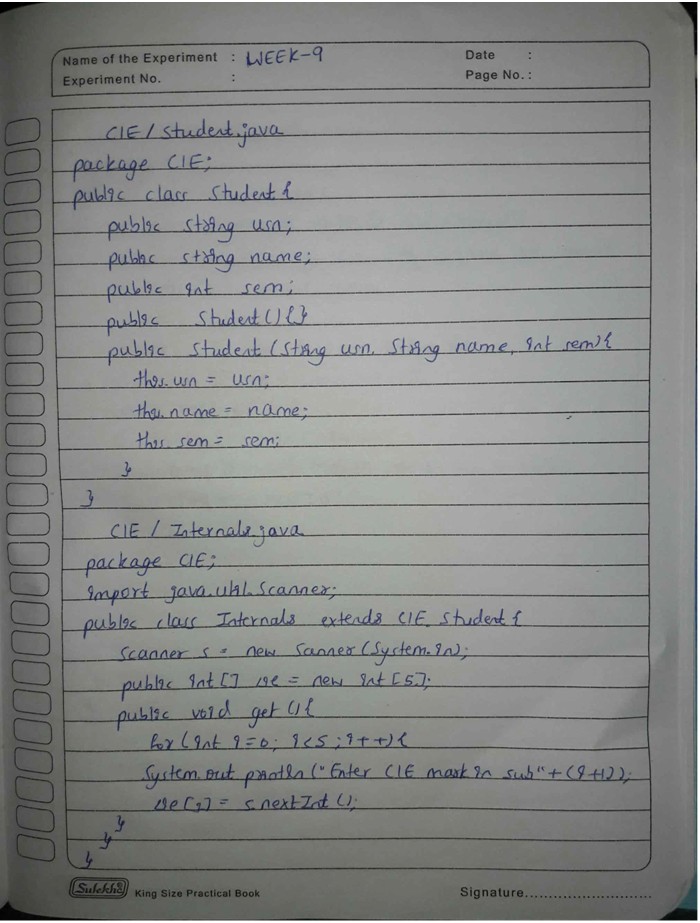
public Student(String usn, String name, int sem) { this.usn = usn;

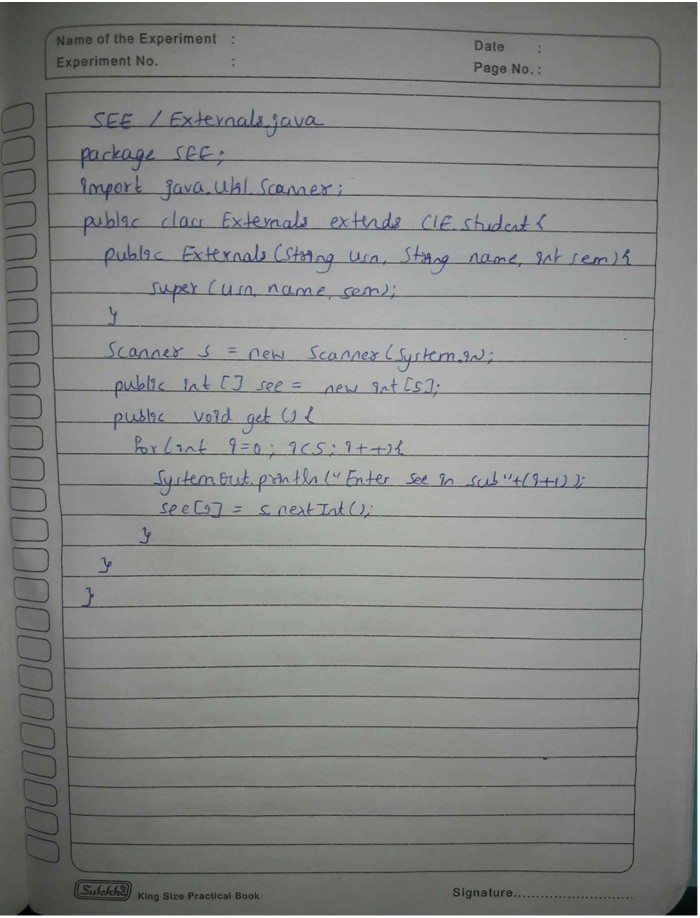
this.name = name; this.sem = sem;

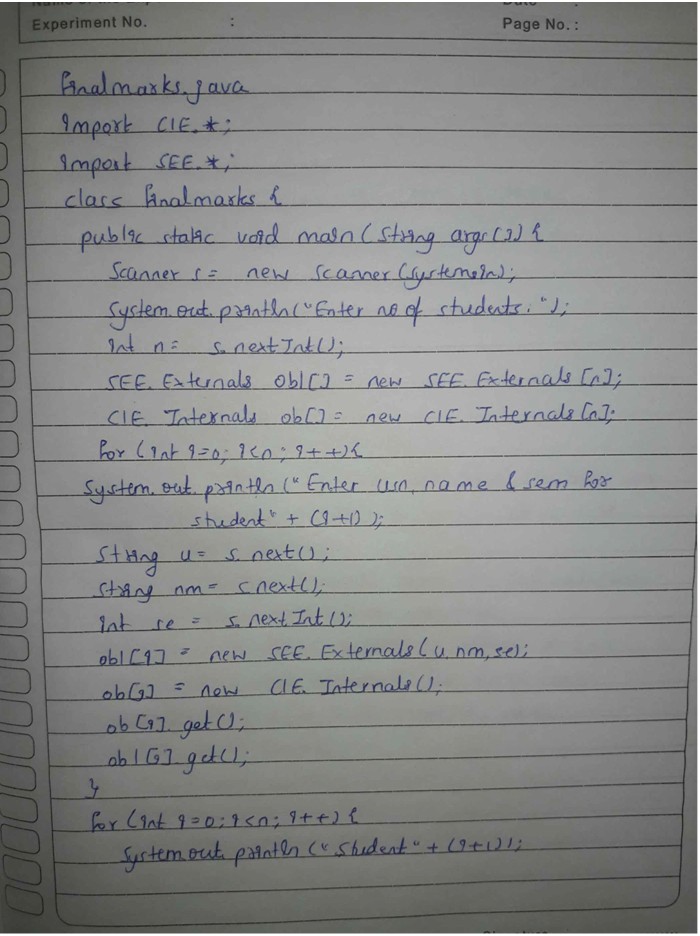
}

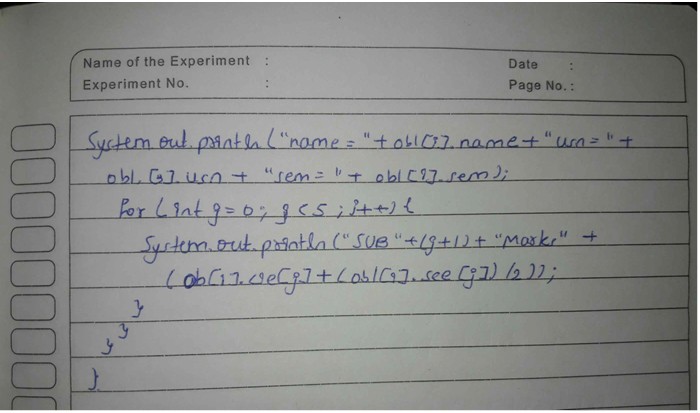
} OBSERVATION:



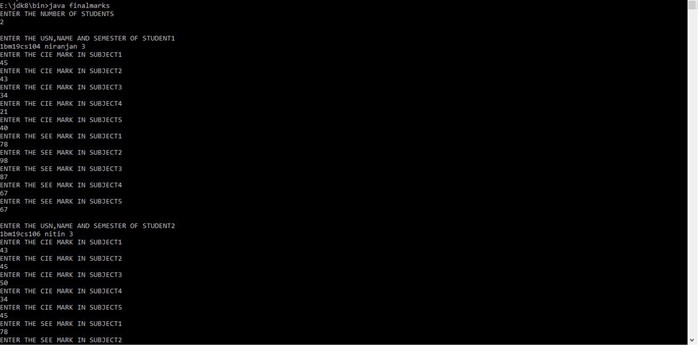








OUTPUT:





# LAB PROGRAM 7

\*Write a program to demonstrate generics with multiple object parameters.

SOURCE CODE:

class Gen<T, U, V> { T ob1;

U ob2; V ob3;

Gen(T o1, U o2, V o3) {

ob1 = o1; ob2 = o2; ob3 = o3;

}

void showTypes() {

System.out.println("Type of T object is " + ob1.getClass().getName());

System.out.println("Type of U object is " + ob2.getClass().getName());

System.out.println("Type of V object is " + ob3.getClass().getName());

}

T getob1() { return ob1;

}

U getob2() { return ob2;

}

V getob3() { return ob3;

}

}

class Generics {

public static void main(String args[]) {

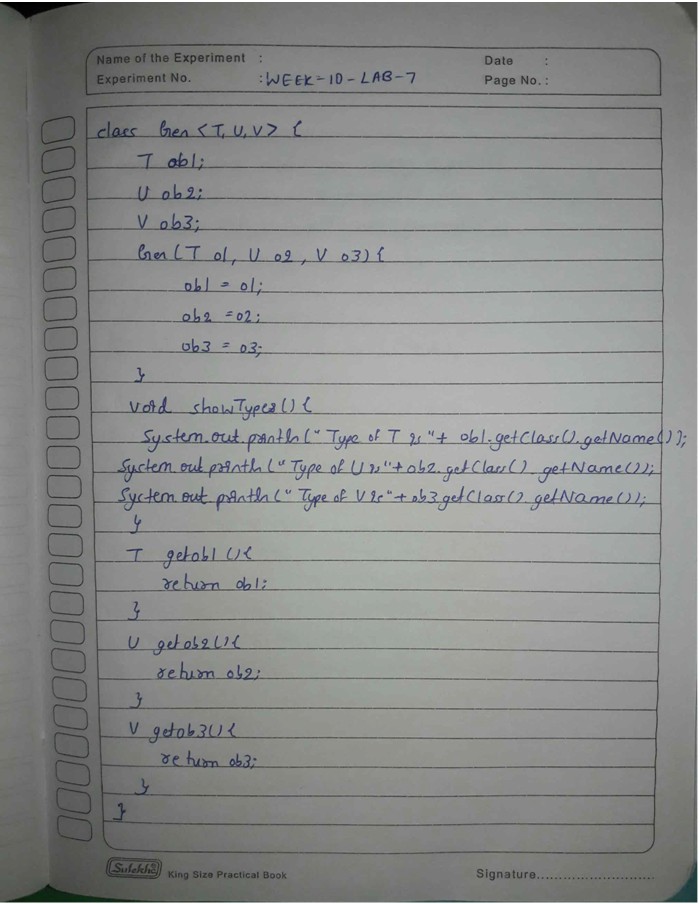
Gen<Integer, Double, String> genOb = new Gen<Integer, Double, String>(15, 99.457, "Niranjan");

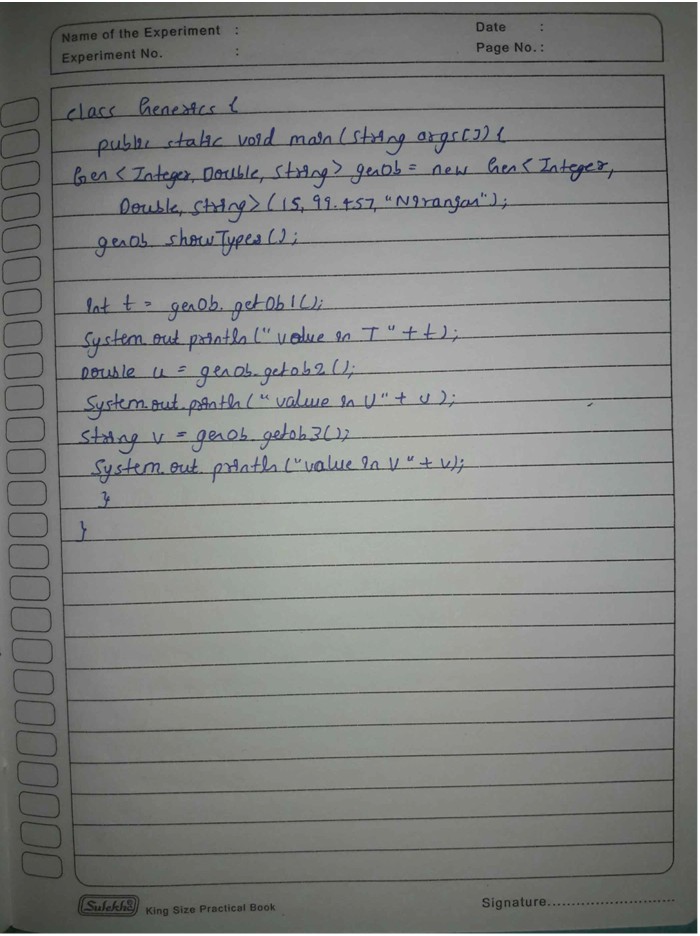
genOb.showTypes();

int t = genOb.getob1(); System.out.println("value in T: " + t); Double u = genOb.getob2(); System.out.println("value in U: " + u); String v = genOb.getob3(); System.out.println("value in V: " + v);

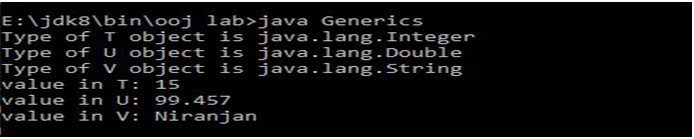
}

} OBSERVATION:





OUTPUT:



# 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 Wrong Age( ) 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>=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 FATHER=" + age + " IS ENTERED INCORRECTLY";

}

}

class WrongAgeSon extends Exception {

int age; WrongAgeSon(int x) { age = x;

}

public String toString() {

return "AGE OF SON=" + age + " IS ENTERED INCORRECTLY";

}

}

class Father { int a; Father(int x) { a = x;

}

void check() throws WrongAge { if (a<0) {

throw new WrongAge(a);

}

}

}

class Son extends Father { int age;

Son(int fage,int sage) { super(fage);

age = sage;

}

void compute() throws WrongAgeSon { if (age >= a) {

throw new WrongAgeSon(age);

} else {

System.out.println("THE AGES ARE ENTERED CORECTLY"); System.out.println("FATHER'S AGE=" + a + "\t" + "SON'S AGE="

+ age);

}

}

}

class ExceptionsMain1 {

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.check(); try { ss.compute();

} catch (WrongAgeSon e) { System.out.println(e);

}

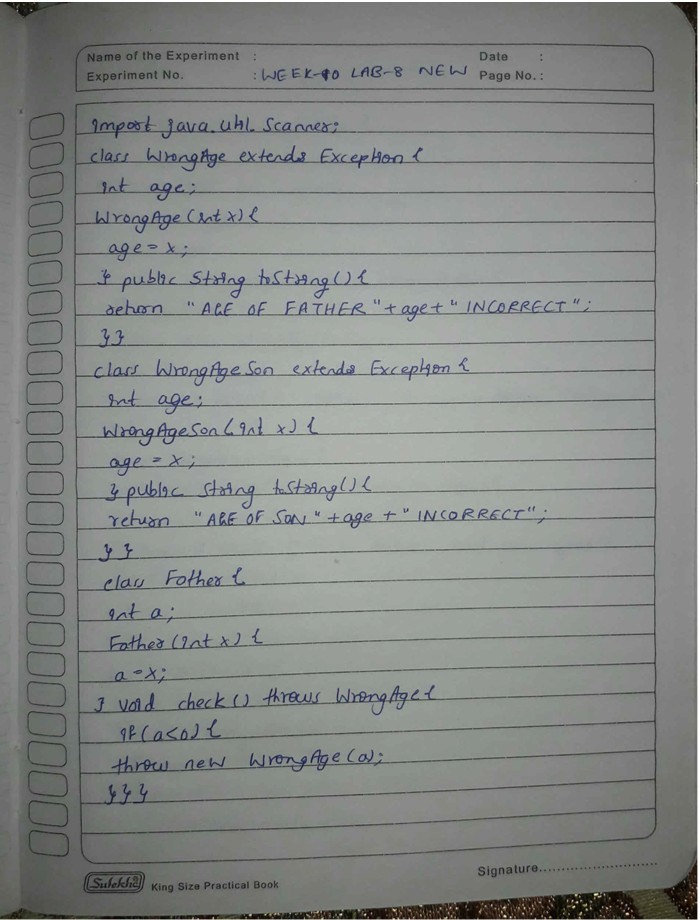
} catch (WrongAge e) { System.out.println(e);

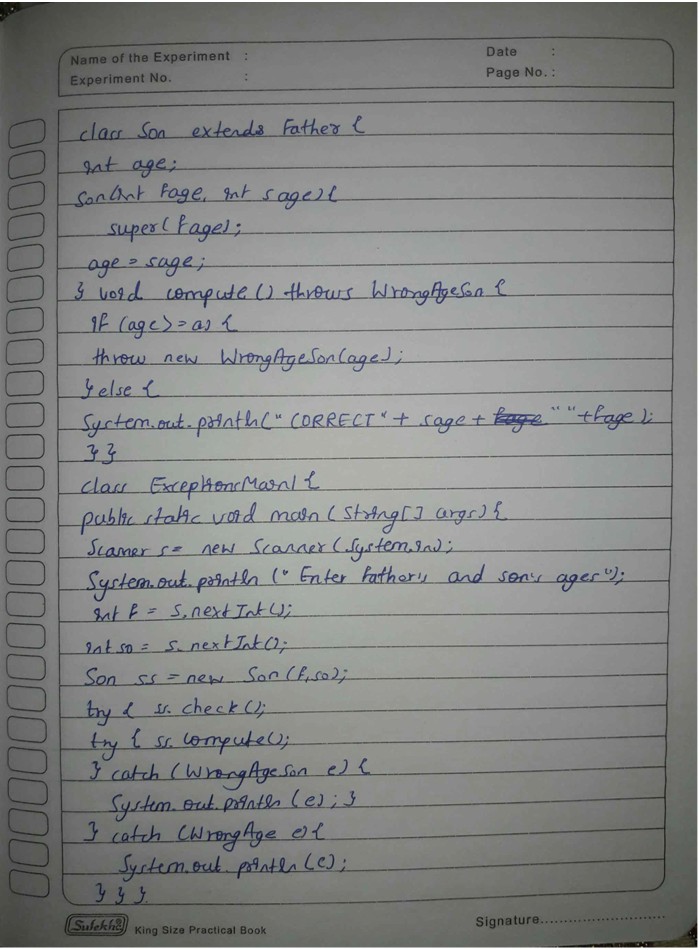
}

}

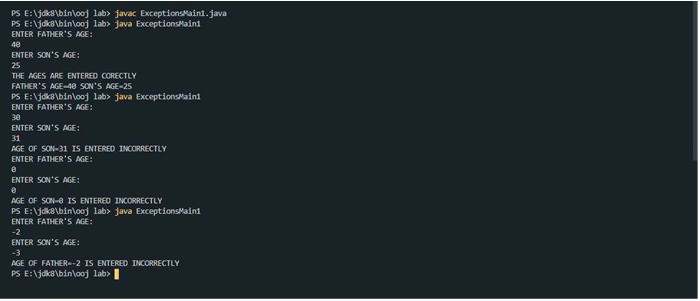
}

OBSERVATION:





OUTPUT:



# 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.

SOURCE CODE:

class Thread1 implements Runnable { String name;

Thread t; int time;

Thread1(String threadname, int time) { name = threadname;

this.time = time;

t = new Thread(this, name); System.out.println("thread:" + t); t.start();

}

public void run() { try {

for (int i = 5; i > 0; i--) { System.out.println(name); Thread.sleep(time);

}

} catch (InterruptedException e) { System.out.println(name + "Interrupted");

}

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

}

}

class ThreadMain1 {

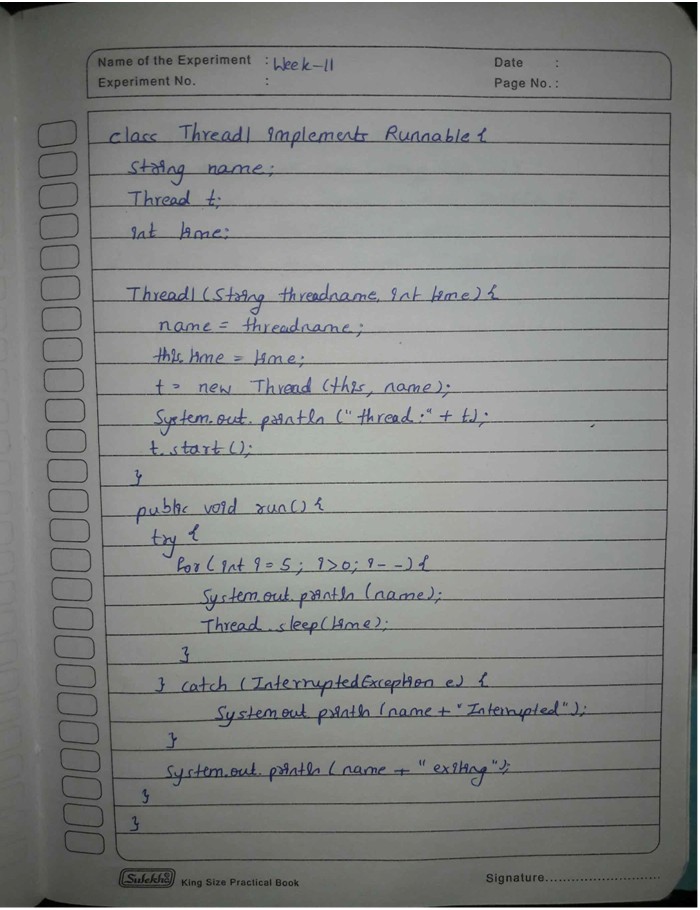
public static void main(String args[]) {

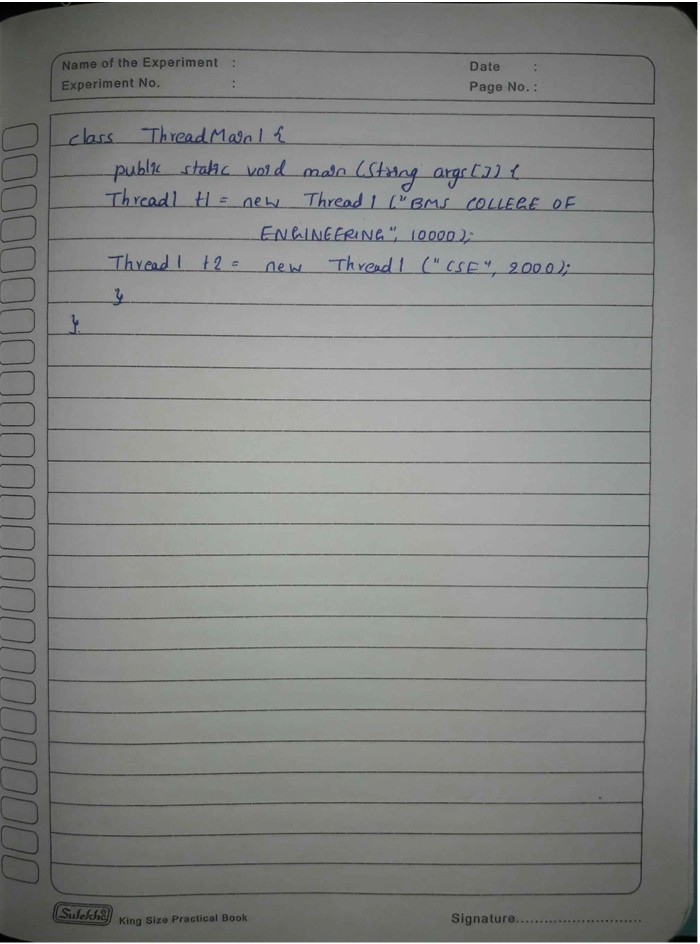
Thread1 t1 = new Thread1("BMS COLLEGE OF ENGINEERING", 10000);

Thread1 t2 = new Thread1("CSE", 2000);

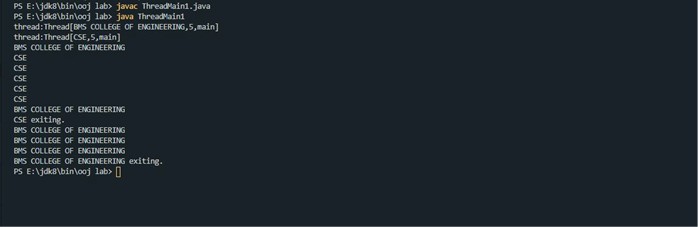
}

} OBSERVATION:





OUTPUT:



# 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.\*;

class dia extends Dialog implements ActionListener { integerdivision1 id;

dia(Frame parent, String title) { super(parent, title, false);

id = (integerdivision1) parent; setLayout(new FlowLayout()); setSize(300, 200);

add(new Label(id.msg)); Button b;

add(b = new Button("OK")); b.addActionListener(this);

}

public void actionPerformed(ActionEvent ae) { dispose();

}

}

public class integerdivision1 extends Frame implements ActionListener {

String msg = ""; TextField n1, n2, res; Label ln1, ln2, lres; Button b;

public integerdivision1() { 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 ){

msg="NUMBERFORMAT EXCEPTION";

dia d=new dia(this,"EXCEPTION"); d.setVisible(true);

}

catch(ArithmeticException a){

msg="ARITHMETIC EXCEPTION";

dia d=new dia(this,"EXCEPTION");

d.setVisible(true);

}

}

}

public static void main(String args[]) { integerdivision1 i = new integerdivision1(); i.setSize(new Dimension(300, 300));

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

i.setVisible(true);

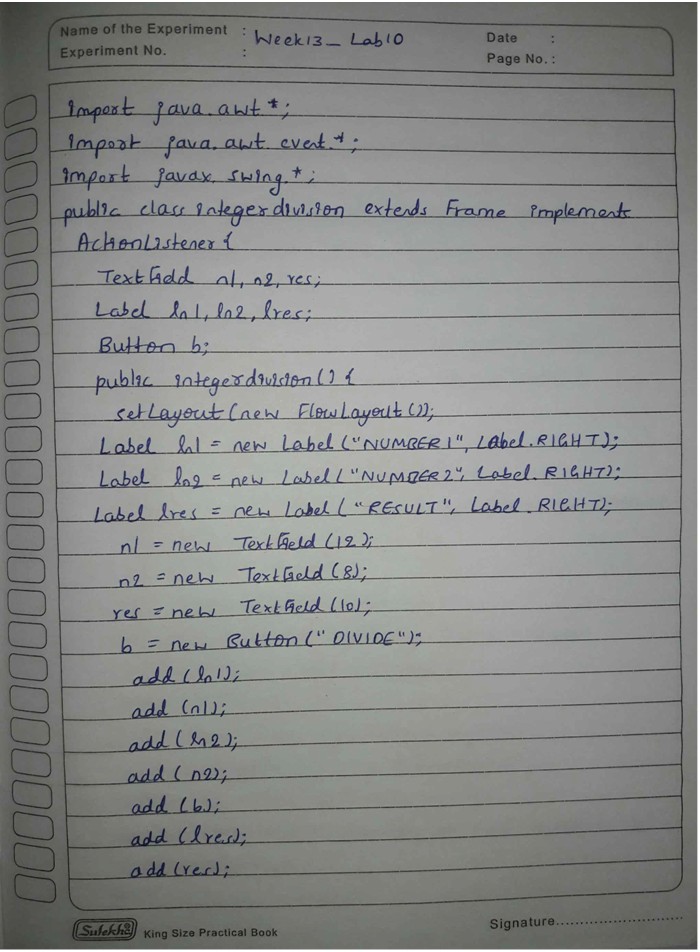
}

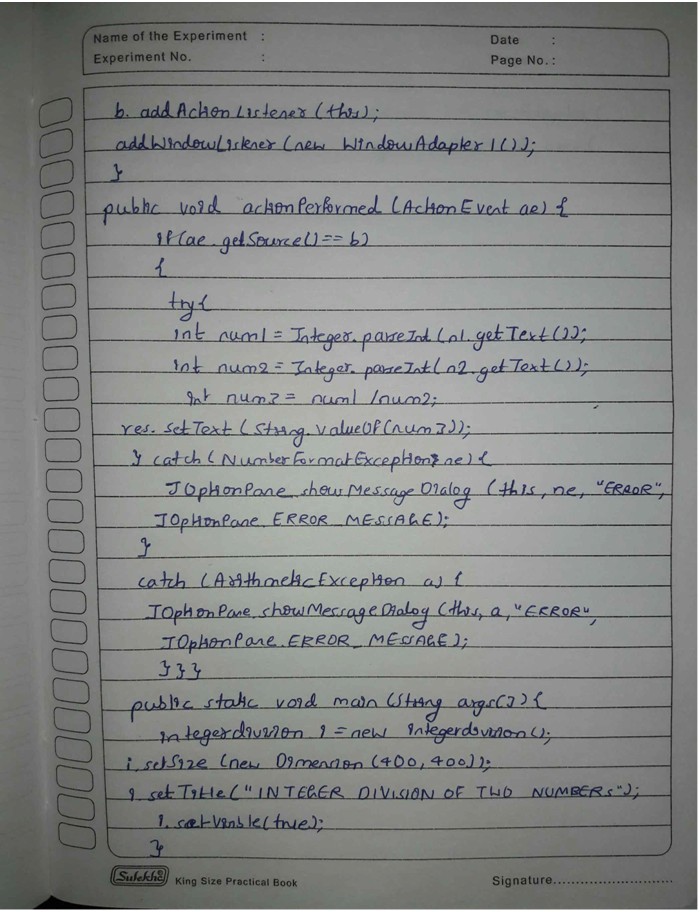
class WindowAdapter1 extends WindowAdapter { public void windowClosing(WindowEvent we) { System.exit(0);

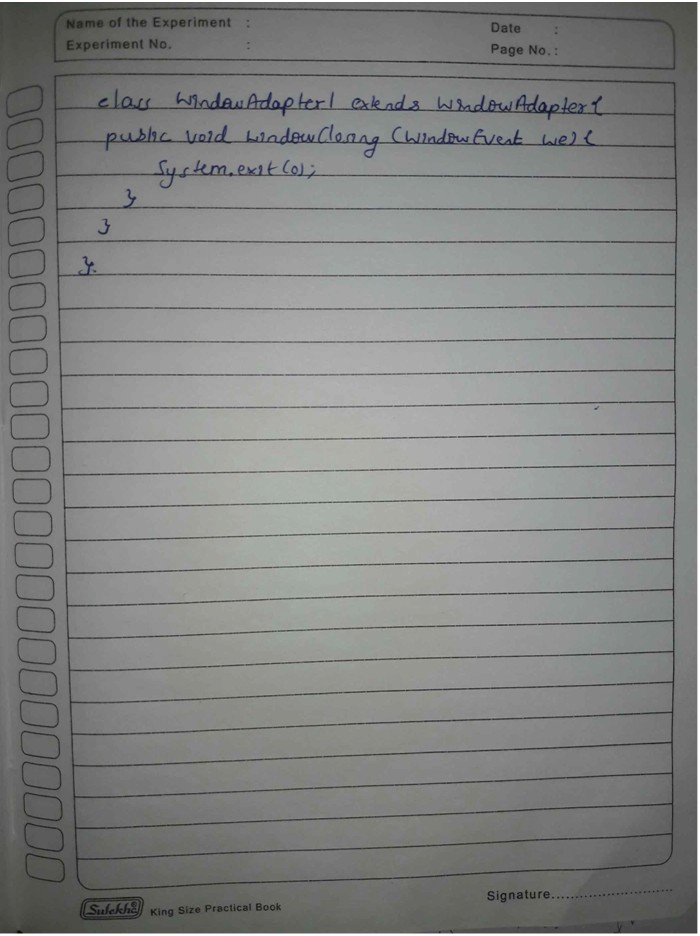
}

}

} OBSERVATION:







OUTPUT:





