1. 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.

Student.java

**package** CIE;

**public** **class** Student {

**public** **int** usn;

**public** String name;

**public** **int** sem;

}

Internals.java

**package** CIE;

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

**public** **class** Internals {

**public** **int** internals[]=**new** **int**[5];

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

**public** **void** GetDetails() {

System.***out***.println("Enter marks of CIE : \n ");

**for**(**int** j=1;j<=5;j++) {

System.***out***.println("Enter marks of course : " + j);

internals[j-1]=input.nextInt();

}

}

}

Externals.java

package SEE;

import java.util.Scanner;

import CIE.\*;

public class externals extends Student {

public int semester[] =new int[5];

Scanner in = new Scanner(System.in);

public void EnterDetails(int i) {

System.out.println("Student " + i);

System.out.println("Enter the usn of student:\n");

usn=in.nextInt();

System.out.println("Enter the name of student:\n");

name=in.next();

System.out.println("Enter the semester of student:\n");

sem=in.nextInt();

System.out.println("Enter marks of SEE : \n ");

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

System.out.println("Enter marks of course " + (j+1));

semester[j]=in.nextInt();

}

}

}

Marks.java

package marks;

import CIE.\*;

import SEE.\*;

import java.util.\*;

class marks {

public static void main(String args[]) {

int n;

Scanner in = new Scanner(System.in);

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

n=in.nextInt();

Internals[] cie=new Internals[n];

externals[] sem=new externals[n];

//externals semMarks=new externals();

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

sem[i]=new externals();

sem[i].EnterDetails(i+1);

cie[i]=new Internals();

cie[i].GetDetails();

}

System.out.println("The Marks And Details of Students Registered are:\n");

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

System.out.println("Student " + (j+1));

System.out.println("USN: "+sem[j].usn);

System.out.println("Name: "+sem[j].name);

System.out.println("Semester: "+sem[j].sem);

System.out.println("Final Marks :\n");

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

System.out.println("Course " + (i+1) + "\n"+((sem[j].semester[i]/2)+cie[j].internals[i]));

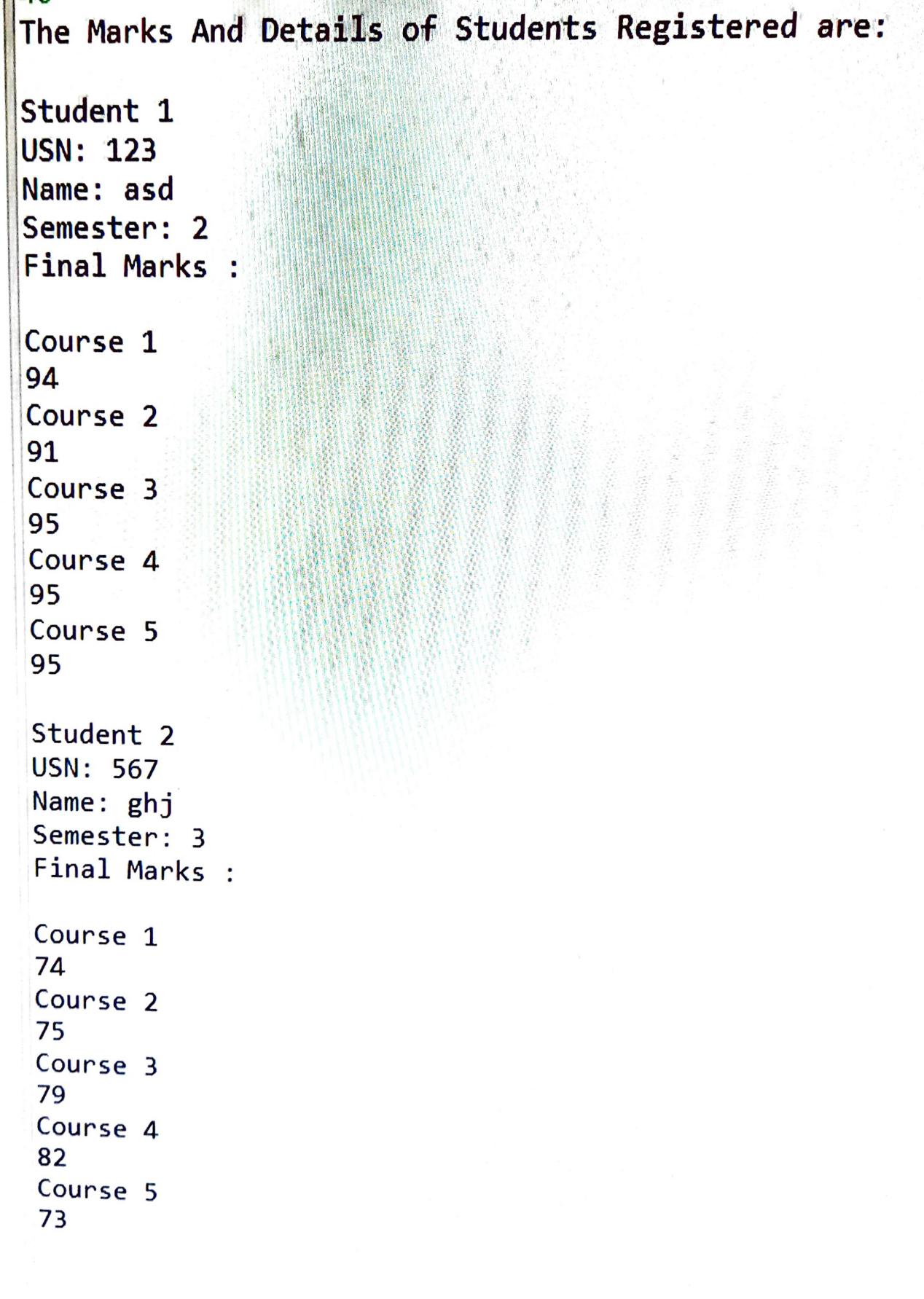
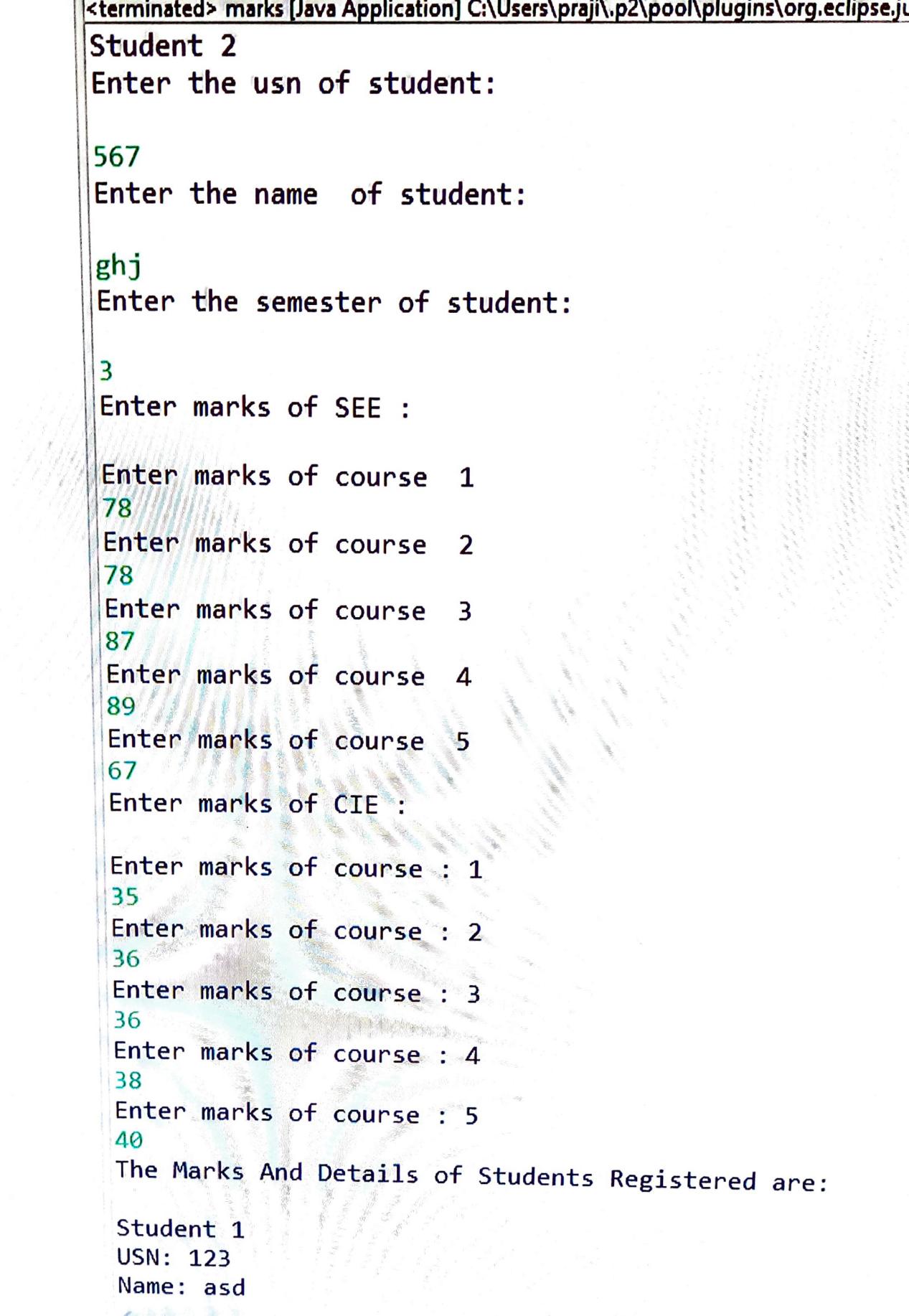
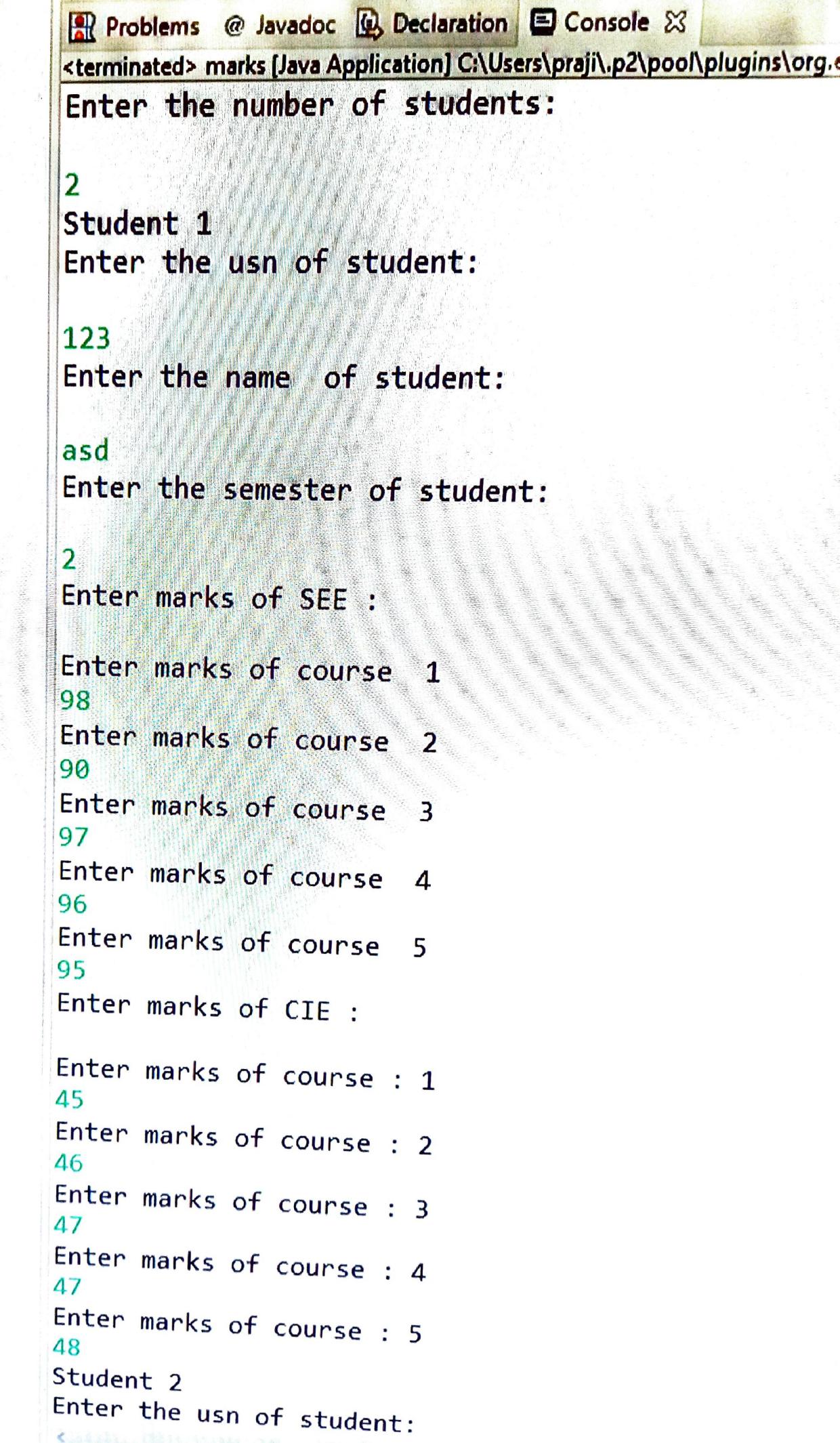
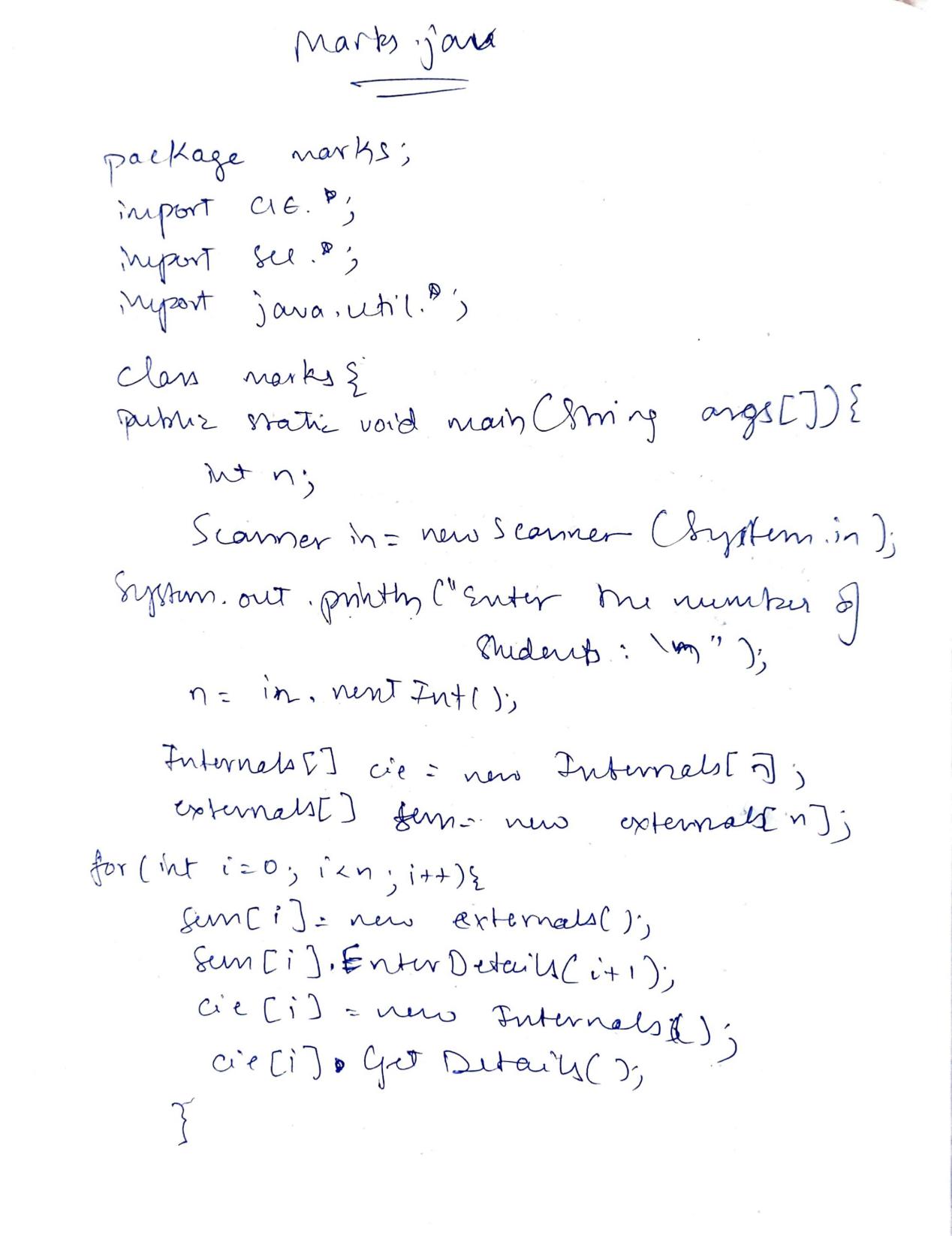
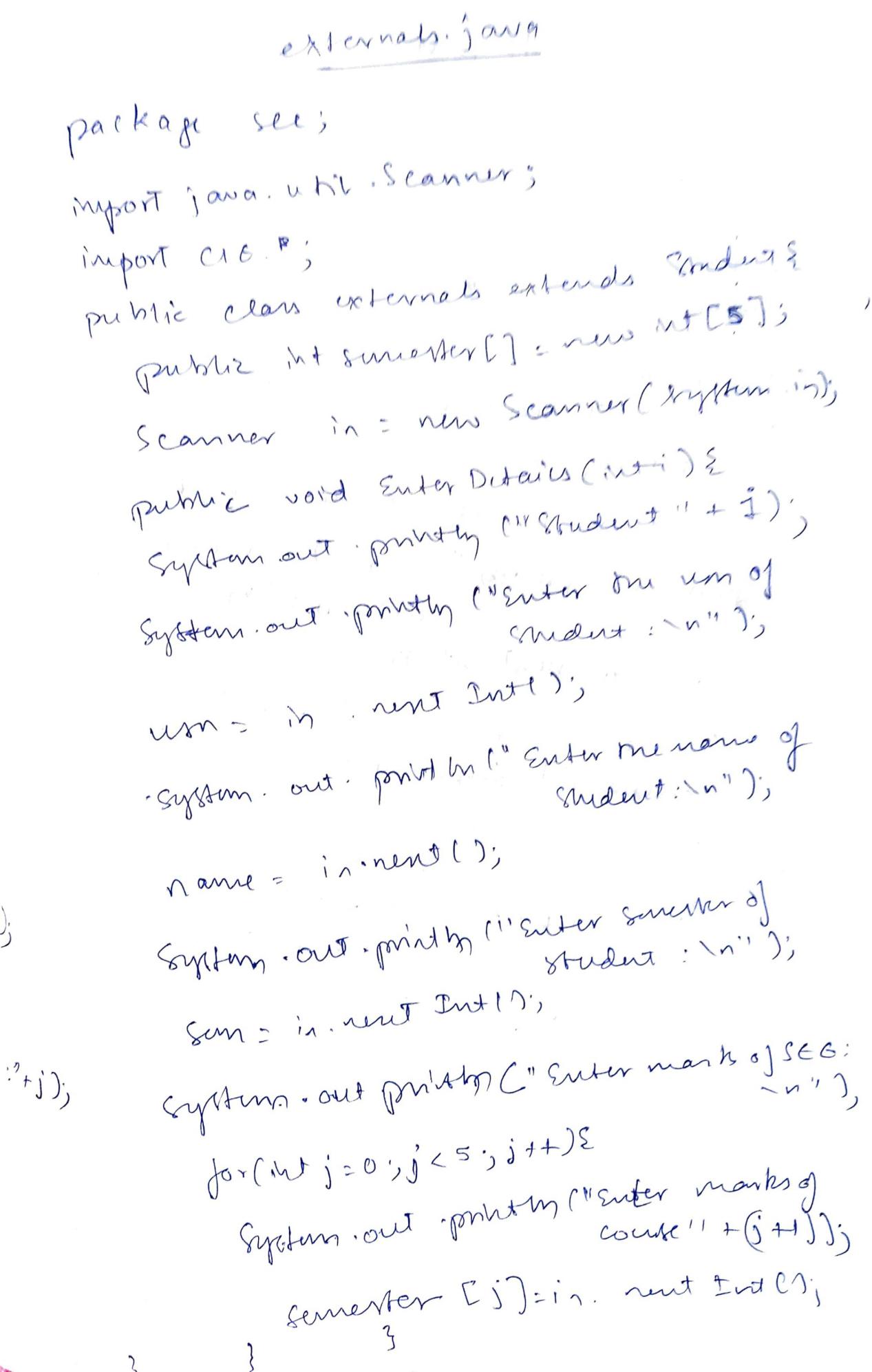
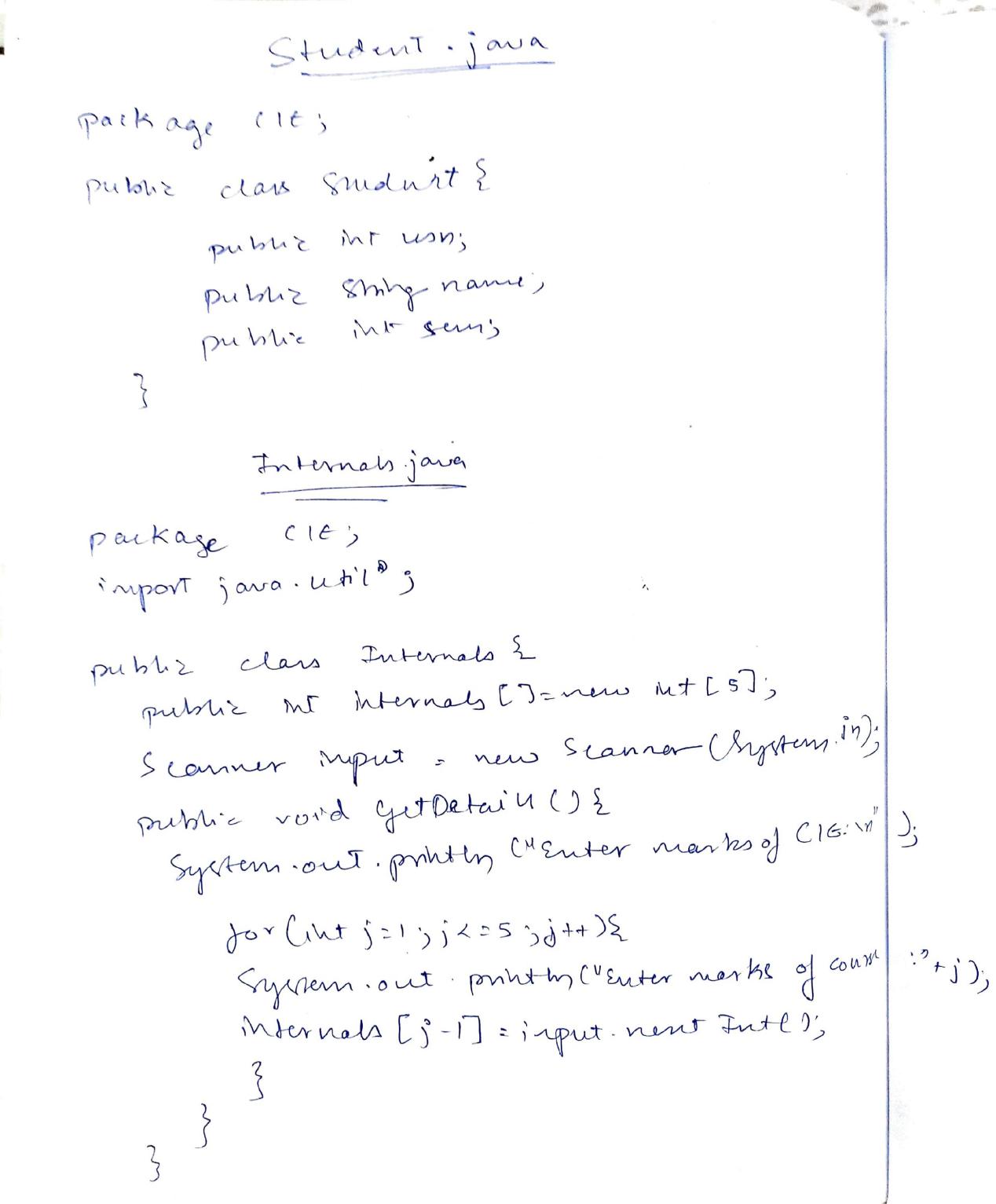
}

System.out.println();

}

}

}



1. Write a program to demonstrate generics with multiple object parameters.

**class** Gen<I> {

I ob1;

I ob2;

Gen(I o1, I o2) {

ob1 = o1;

ob2 = o2;

}

I a() {

**return** ob1 ;

}

I b() {

**return** ob2;

}

}

**class** Generics {

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

Gen<Integer> i = **new** Gen<Integer>(88, 90);

Gen<Double> f = **new** Gen<Double>(88.2234,90.1234);

Gen<String> s = **new** Gen<String>("88","90");

**int** a = i.a();

**int** b=i.b();

System.***out***.println(" INT value: " + (a+b));

**double** x = f.a();

**double** y=f.b();

System.***out***.println(" Double value: " + (x+y));

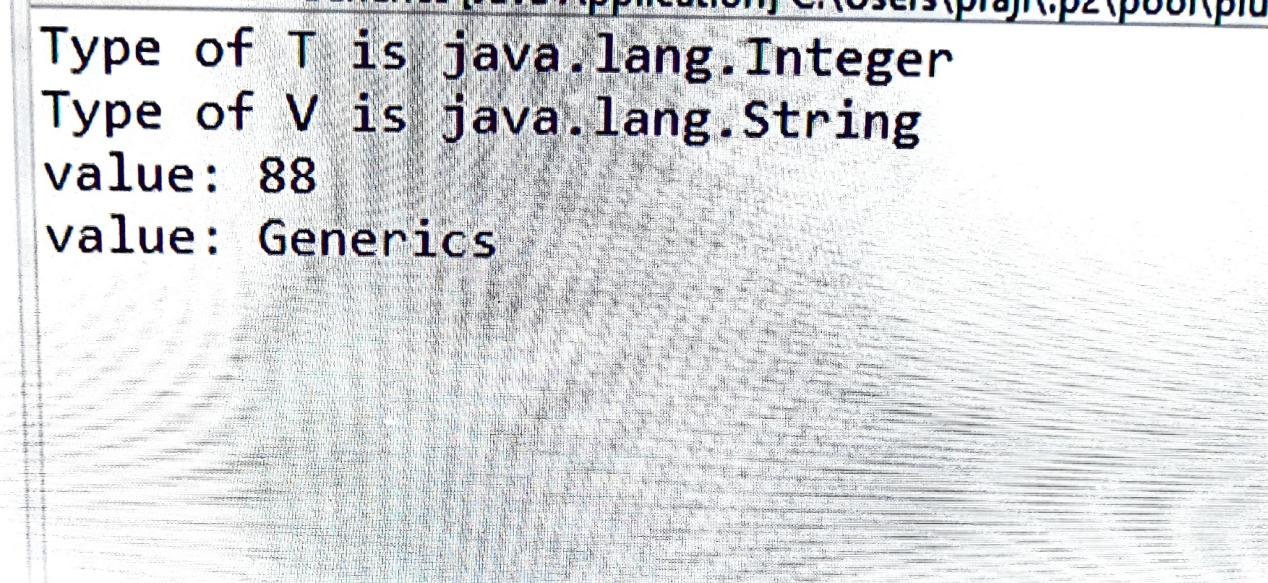
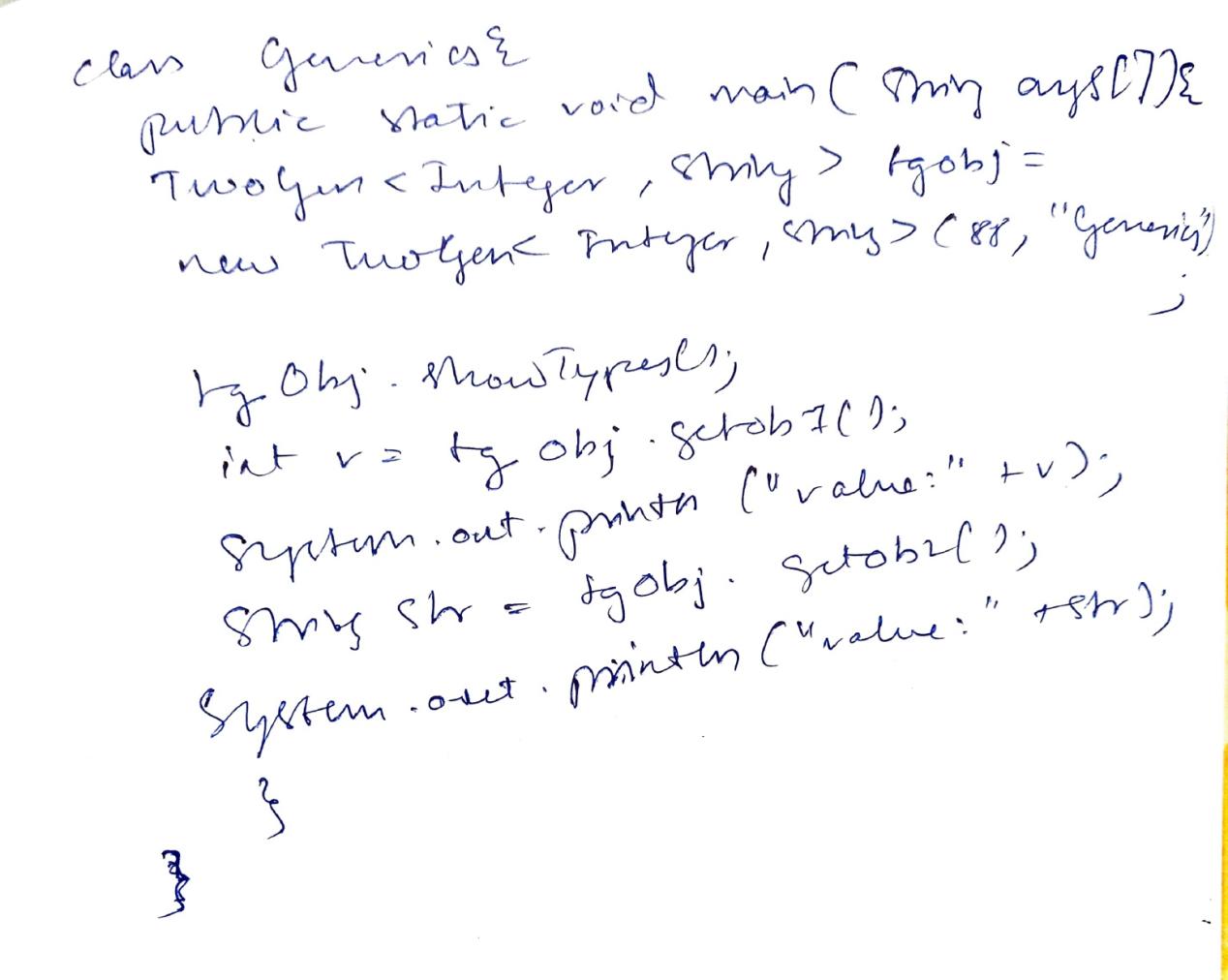
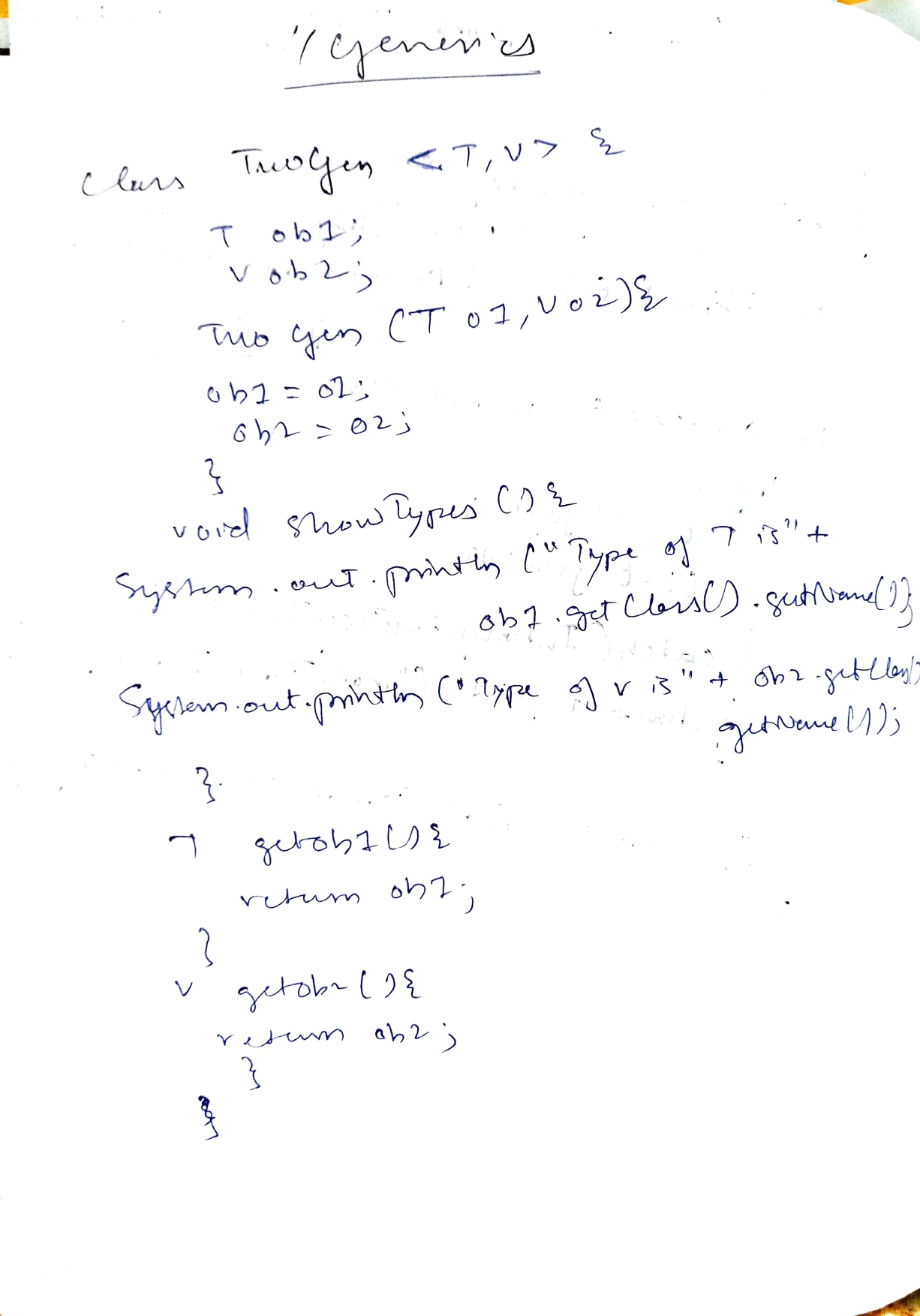
String s1=s.a();

String s2=s.b();

System.***out***.println(" String value: " + (s1+s2));

}

}



1. 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.

**package** interface\_exceptionhandling;

**class** Father **extends** Exception {

**private** **int** Fage;

Father(){}

Father(**int** a) {

Fage=a;

}

**public** String toString() {

**return** "Father Age cannot be negative....";

}

}

**class** Son **extends** Father{

**int** sage;

**int** fage;

Son(**int** s,**int** f){

sage=s;

fage=f;

}

**public** String toString() {

**return** "Father Age cannot be less than or equal son Age....";

}

}

**class** exception {

**static** **void** AgeCompare(**int** fage,**int** sage) **throws** Son {

**if**(fage<=sage)

**throw** **new** Son(sage,fage);

}

**static** **void** NegativeAgeCheck(**int** age) **throws** Father {

**if**(age<0)

**throw** **new** Father(age);

}

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

**try** {

*NegativeAgeCheck*(-30);

}**catch**(Father e){

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

}

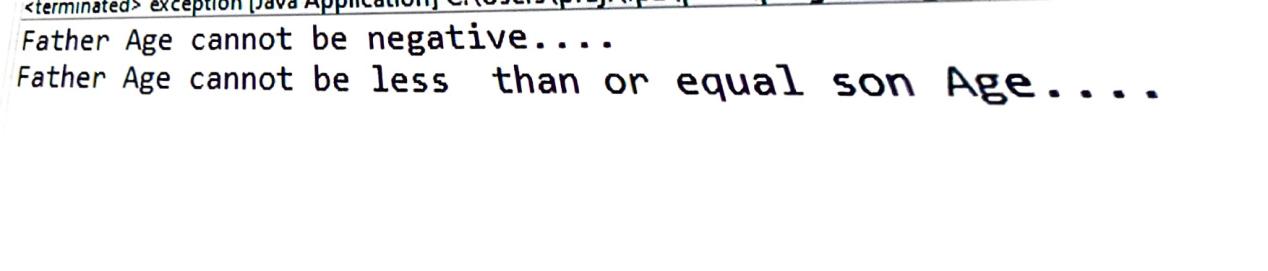
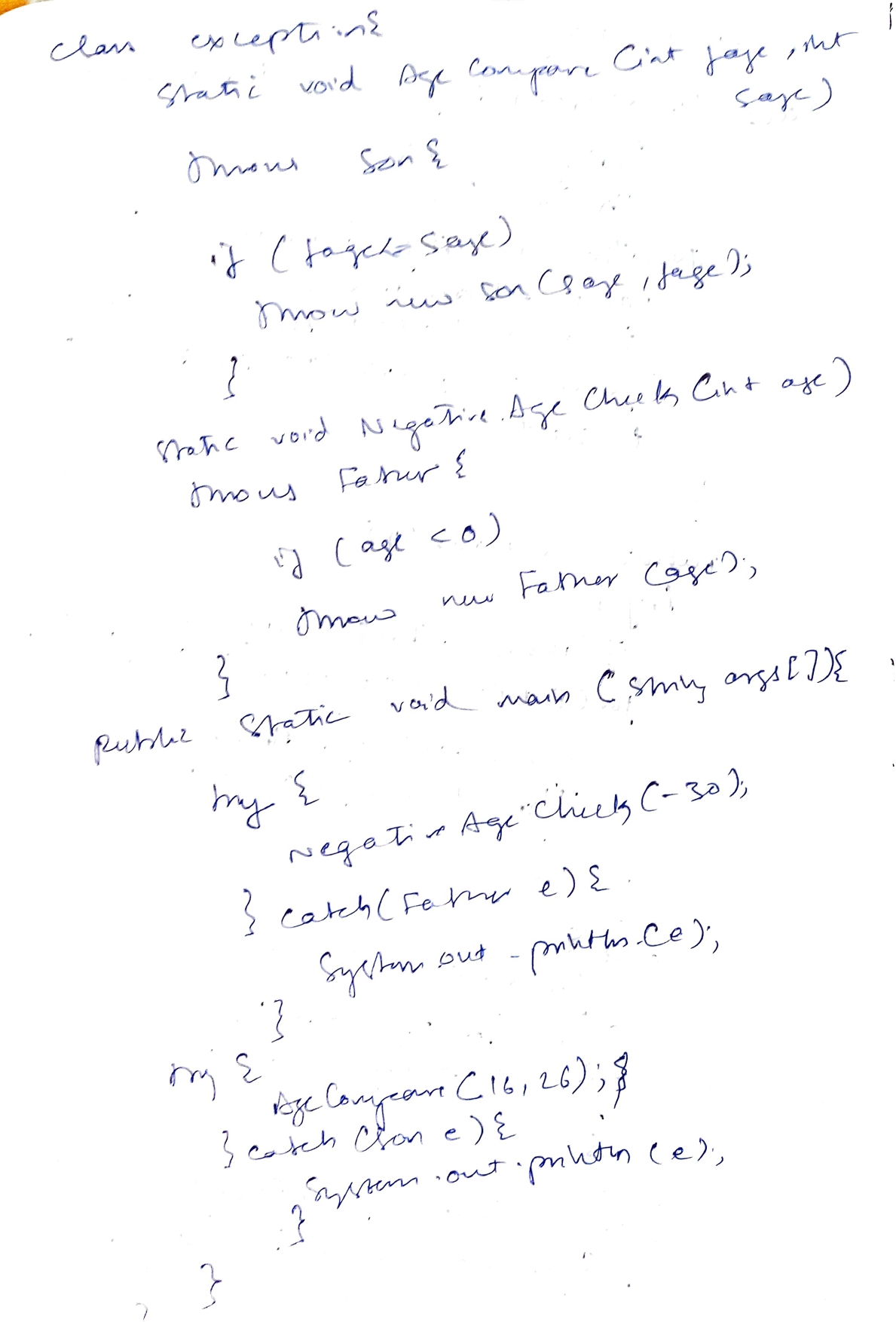
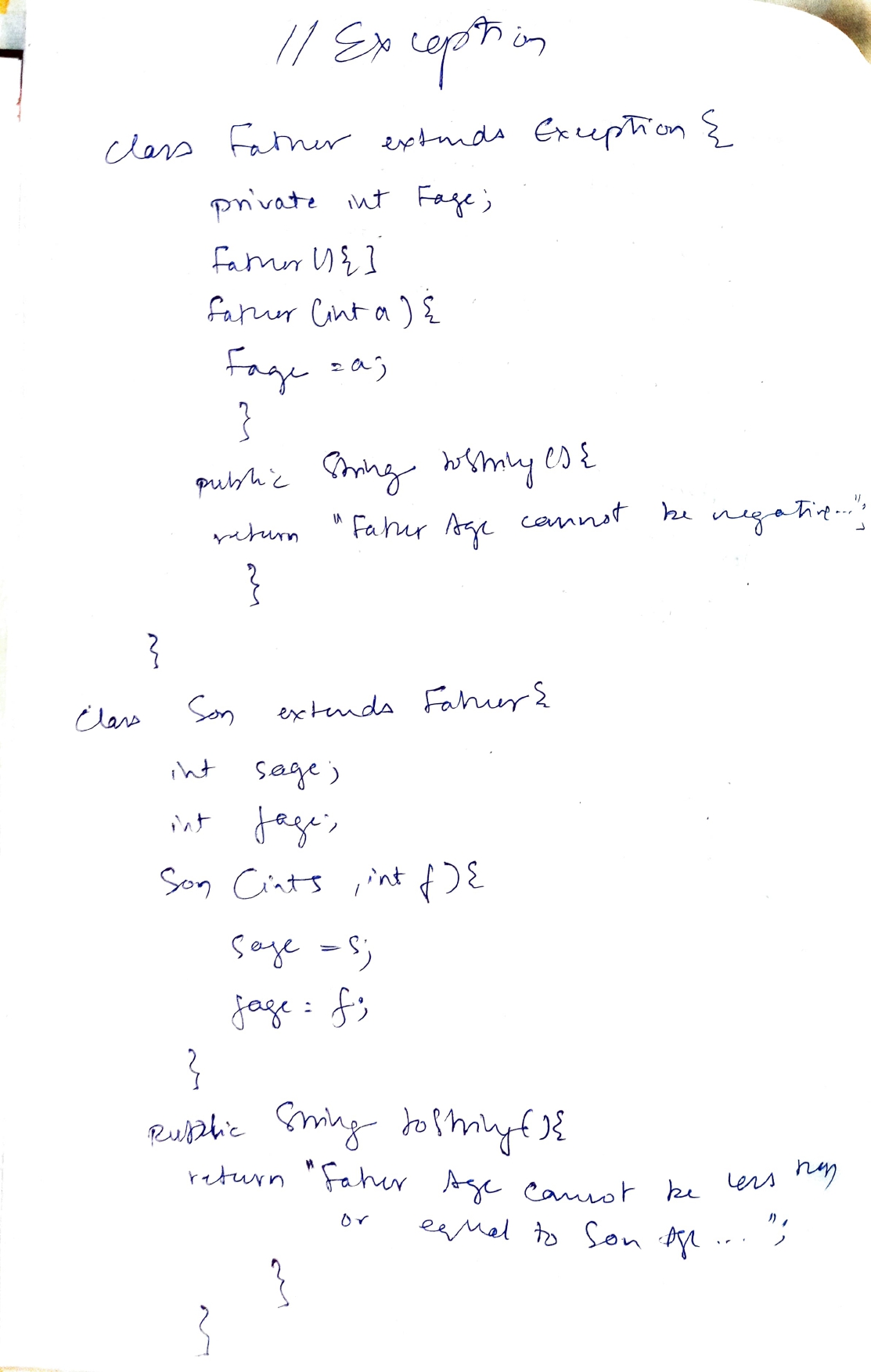
**try**{*AgeCompare*(16,26);}**catch**(Son e){

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

}

}

}



1. 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.

**class** NewThread **implements** Runnable{

String name;

Thread t;

NewThread(String threadname){

name=threadname;

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

System.***out***.println("Child thread : " + t);

t.start();

}

**public** **void** run() {

**try** {

**if**(name=="BMS College of Engineering") {

**for**(**int** i=0;i<3;i++) {

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

Thread.*sleep*(10000);

}

}

**else**{

**for**(**int** i=0;i<15;i++) {

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

Thread.*sleep*(2000);

}

}

}**catch**(InterruptedException e) {

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

}

System.***out***.println("Exiting child thread.");

}

}

**public** **class** threading {

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

NewThread t1= **new** NewThread("BMS College of Engineering");

NewThread t2= **new** NewThread("CSE");

**try** {

t1.t.join();

t2.t.join();

}**catch**(InterruptedException e) {

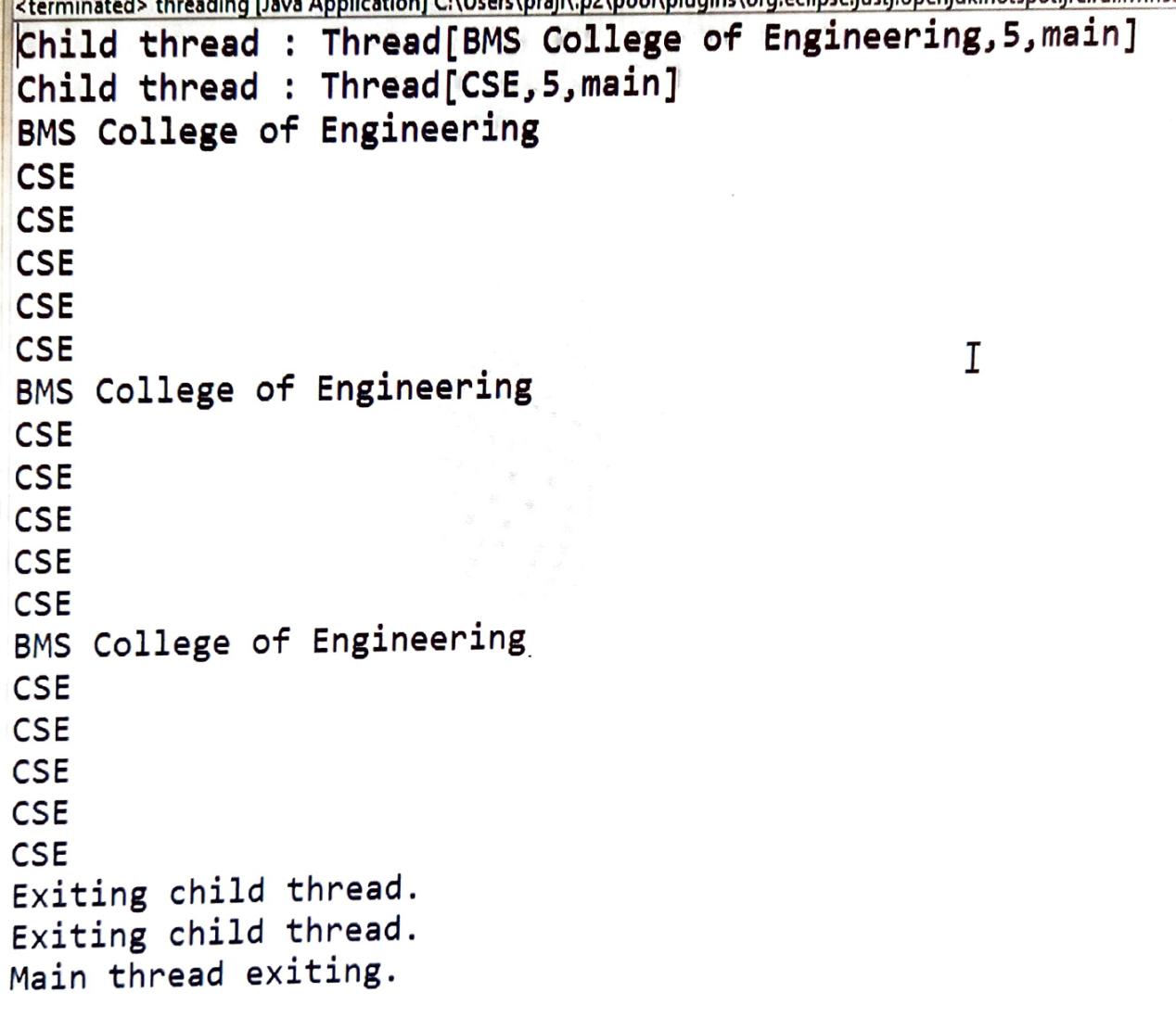
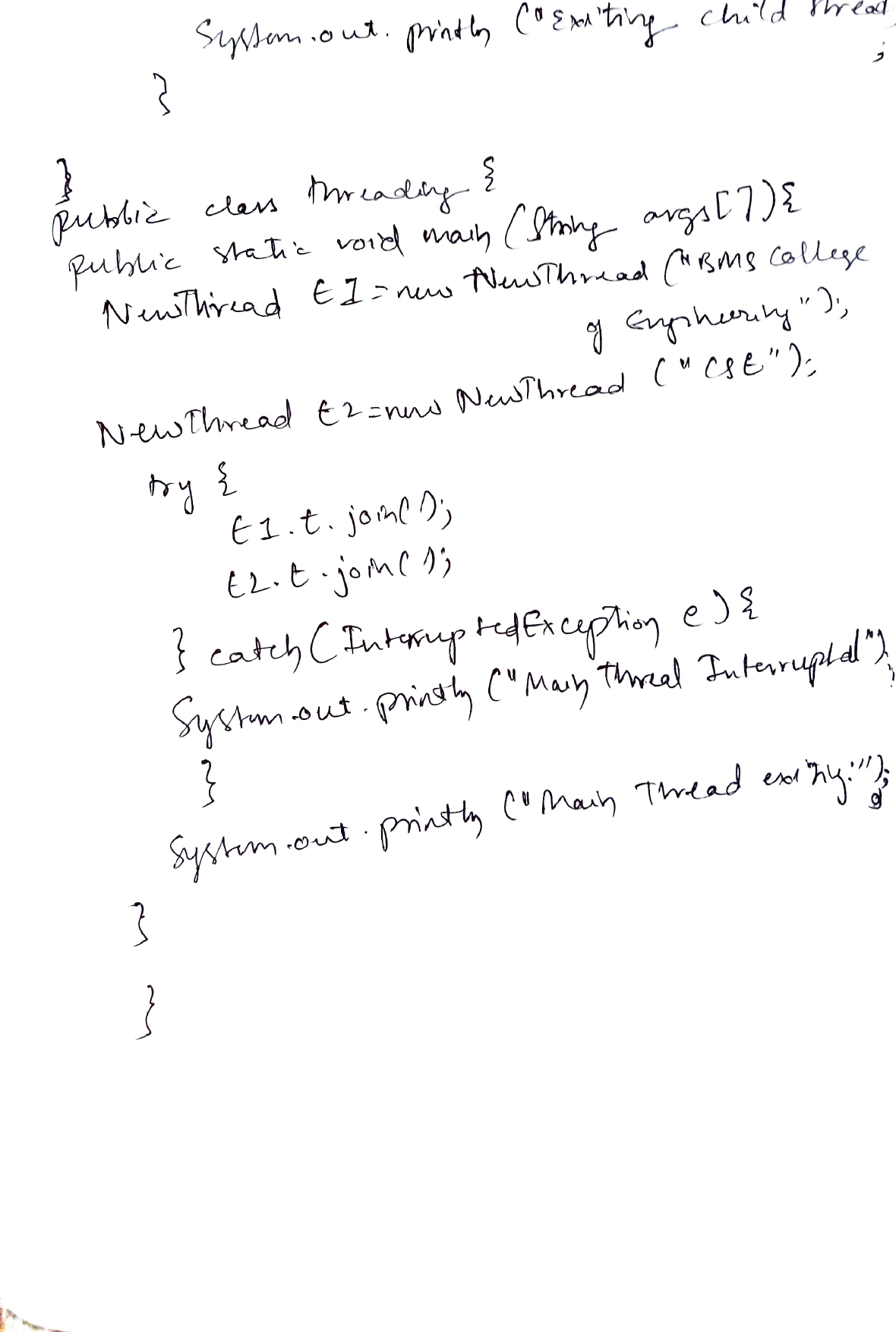
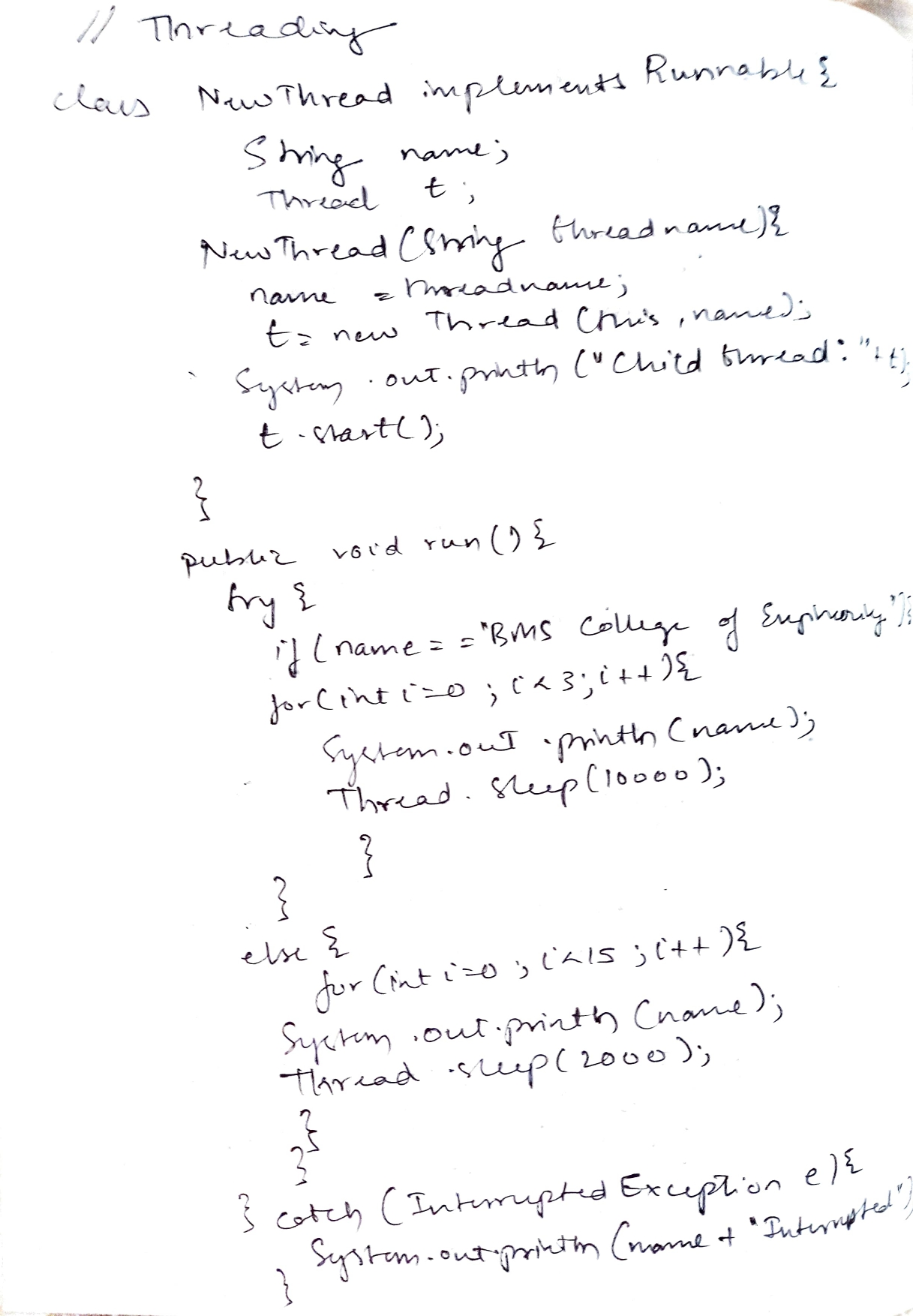
System.***out***.println(" Main Thread Interrupted");

}

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

}

}



1. 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.

import java.awt.\*;

import java.awt.event.\*;

class NumException extends Exception{

public String toString()

{

return "There is an Arithmetic Exception.";

}

}

class FormatException extends Exception{

public String toString()

{

return "There is an Format Exception.";

}

}

public class dividenums extends Frame implements ActionListener{

TextField num1,num2;

Button div;

double result;

String msg="The result is: 0.0";

public dividenums()

{

setLayout(new FlowLayout());

Label num1n= new Label("Numerator: ",Label.RIGHT);

Label num2n= new Label("Denominator: ",Label.RIGHT);

Button div= new Button("Divide");

num1=new TextField(5);

num2=new TextField(5);

add(num1n);

add(num1);

add(num2n);

add(num2);

add(div);

num1.addActionListener(this);

num2.addActionListener(this);

div.addActionListener(this);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent we)

{

System.exit(0);

}

});

}

public boolean isDouble(double num)

{

double dec;

dec=num-(int)num;

if(dec==0.0)

return false;

else

return true;

}

public double divide(double a, double b) throws NumException, FormatException

{

if(b==0.0)

{

throw new NumException();

}

else if(isDouble(a) || isDouble(b))

{

throw new FormatException();

}

return (double) a/b;

}

public void actionPerformed(ActionEvent ae)

{

double a,b;

a=Double.parseDouble(num1.getText());

b=Double.parseDouble(num2.getText());

try{

result=divide(a,b);

msg=("The result is: "+result);

}

catch(NumException ne)

{

msg=ne.toString();

}

catch(FormatException fe)

{

msg=fe.toString();

}

repaint();

}

public void paint(Graphics g)

{

ResultDialog d=new ResultDialog(this, "Result");

d.setVisible(true);

}

public static void main(String args[])

{

dividenums appwin= new dividenums();

appwin.setSize(new Dimension(350,300));

appwin.setTitle("Divide Two Numbers");

appwin.setVisible(true);

}

}

class ResultDialog extends Dialog implements ActionListener{

dividenums pt;

ResultDialog(Frame parent,String title)

{

super(parent,title,false);

pt=(dividenums)parent;

setLayout(new FlowLayout());

setSize(250,100);

Button b=new Button("OK");

add(new Label(pt.msg));

b.addActionListener(this);

add(b);

addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent we){

dispose();

}

});

}

public void actionPerformed(ActionEvent ae)

{

dispose();

}

}

