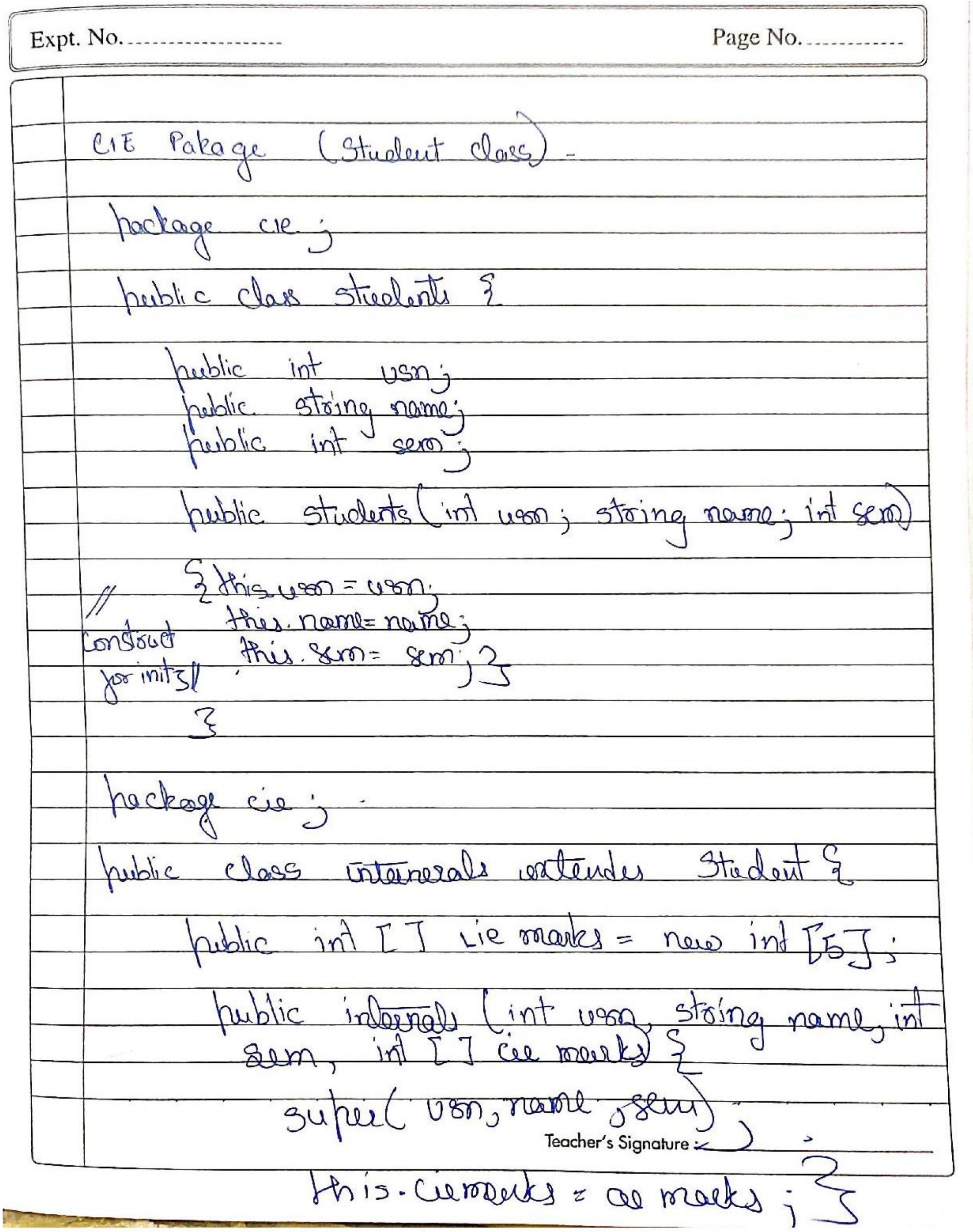
LAB 6:- Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class 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.



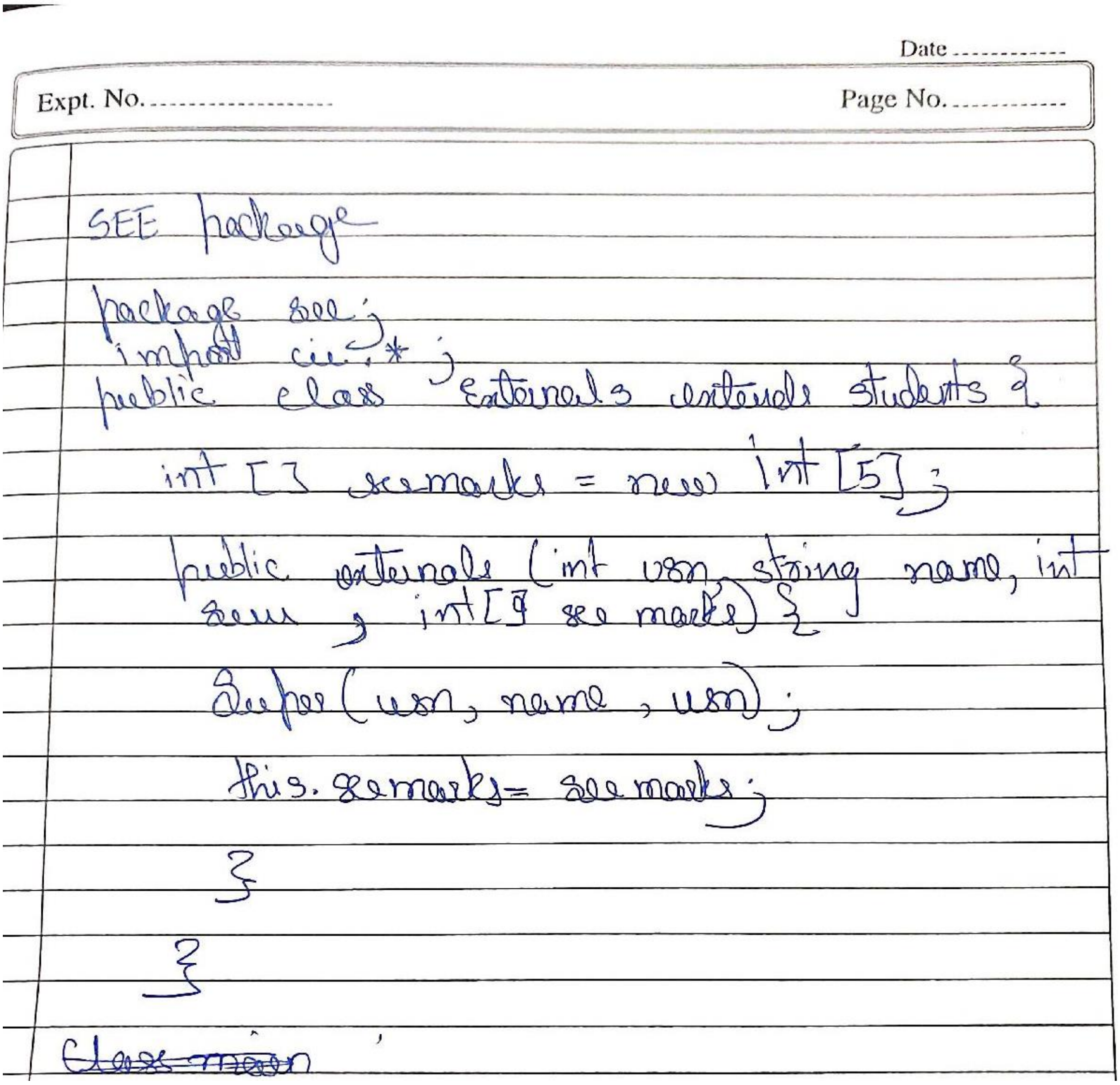
Expt. No. .... Page No. ....

CIE Pakage (Student class) -

```
package cie;
public class students {

    public int usn;
    public string name;
    public int sem;

    public students(int usn; string name; int sem)
//  { this.usn = usn;
Construct  this.name= name;
for init3//  this.sem= sem; }
    }

package cie;
public class interenals extendes Student {
    public int [ ] cie marks = new int [5];
    public internals (int usn, string name, int
     sem, int [ ] cie marks) {
    super( usn, name, sem)
```

Teacher's Signature :

```
    this.ciemarks = cie marks; }
```



Date ....

Expt. No. .... Page No. ....

SEE package

```
package see;
 import cie.*;
public class Externals extends students {
    int [] seemarks = new int [5];
    public externals (int usn, string name, int
     sem , int[] see marks) {
     Super(usn, name , usn);
     this. seemarks= seemarks;
     }
   }
```

~~Class main~~

CODE:-

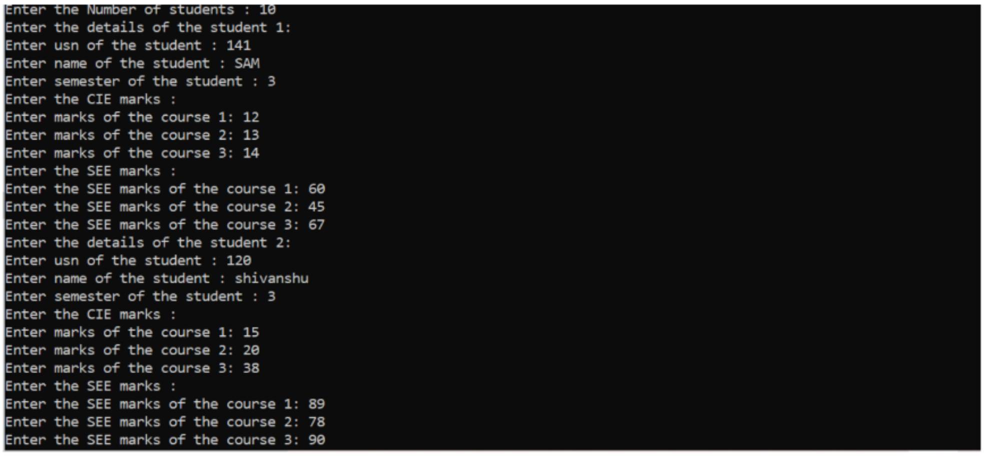
```
    package ciem;
                                          public class CIE {

                                                    public static int usn;
                                                    public static String name;
                                                    public  static int sem;
                                                    public CIE(int usn,String name,int
                                          sem){
                                                       this.usn = usn;
                                                       this.name = name;
                                                       this.sem = sem;
                                                    }
```

```
                                       }
package ciem;
                                public class Internals extends CIE{
                                  public static int[] cieMarks = new int[5];
                                  public Internals(int usn,String name,int sem,int[]
                                cieMarks){
                                     super( usn,name,sem );
                                     this.cieMarks = cieMarks;
                                  }
                                }

                                         package see;
                                         import ciem.*;
                                         public class externalsm extends CIE{
                                                 public static int[] seem= new int [5];
                                                 public externalsm(int usn, String
                                         name,int sem,int [] seem) {
                                                        super(usn,name,usn);
                                                        this.seem=seem;
                                                 }
                                                 }
```

```
package lab6main;
                       import see.*;
                       import ciem.*;
                       import java.util.*;
                       public class Maintotal {
                       public static void main (String [] args ) {
                            Scanner input = new
                       Scanner(System.in);
                            externalsm[] e = new externalsm[2];
                            Internals[] in = new Internals[2];
                            for(int i=0;i<2;i++){
                              int usn1 = input.nextInt();
                              String name1 = input.next();
                              int sem1 = input.nextInt();
                              int[][] cie = new int[2][5];
                              int[][] see = new int[2][5];
                              for(int j=0;j<5;j++){
                                 cie[i][j] = input.nextInt();
                              }
                              for(int j=0;j<5;j++){
                                 see[i][j] = input.nextInt();
```

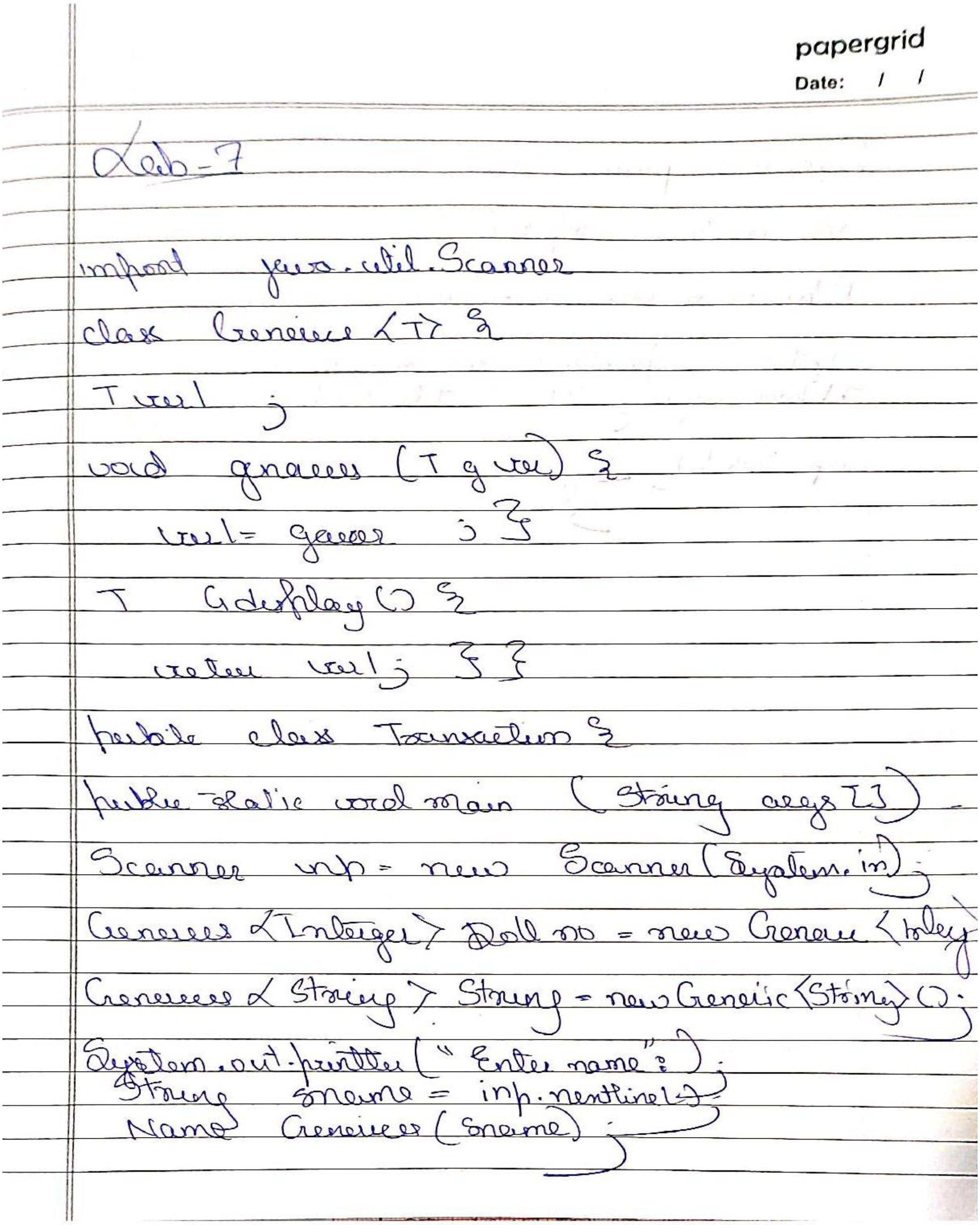
```
            }
            e[i] = new
externalsm(usn1,name1,sem1,see[i]);
            in[i] = new
Internals(usn1,name1,sem1,cie[i]);
            int total = 0;
            System.out.println("Name:
"+e[i].name);
            System.out.println("USN:
"+e[i].usn);
            System.out.println("sem:
"+e[i].sem);
            for(int j=0;j<5;j++){
                total =
e[i].seem[j]+in[i].cieMarks[j];
                System.out.print("Final marks:
"+total+" ");
            }
            System.out.println();
        }
    }
}
```

OUPUT:-



```
Enter the Number of students : 10
Enter the details of the student 1:
Enter usn of the student : 141
Enter name of the student : SAM
Enter semester of the student : 3
Enter the CIE marks :
Enter marks of the course 1: 12
Enter marks of the course 2: 13
Enter marks of the course 3: 14
Enter the SEE marks :
Enter the SEE marks of the course 1: 60
Enter the SEE marks of the course 2: 45
Enter the SEE marks of the course 3: 67
Enter the details of the student 2:
Enter usn of the student : 120
Enter name of the student : shivanshu
Enter semester of the student : 3
Enter the CIE marks :
Enter marks of the course 1: 15
Enter marks of the course 2: 20
Enter marks of the course 3: 38
Enter the SEE marks :
Enter the SEE marks of the course 1: 89
Enter the SEE marks of the course 2: 78
Enter the SEE marks of the course 3: 90
```

LAB 7:-Write a program to demonstrate generics with multiple object parameters.

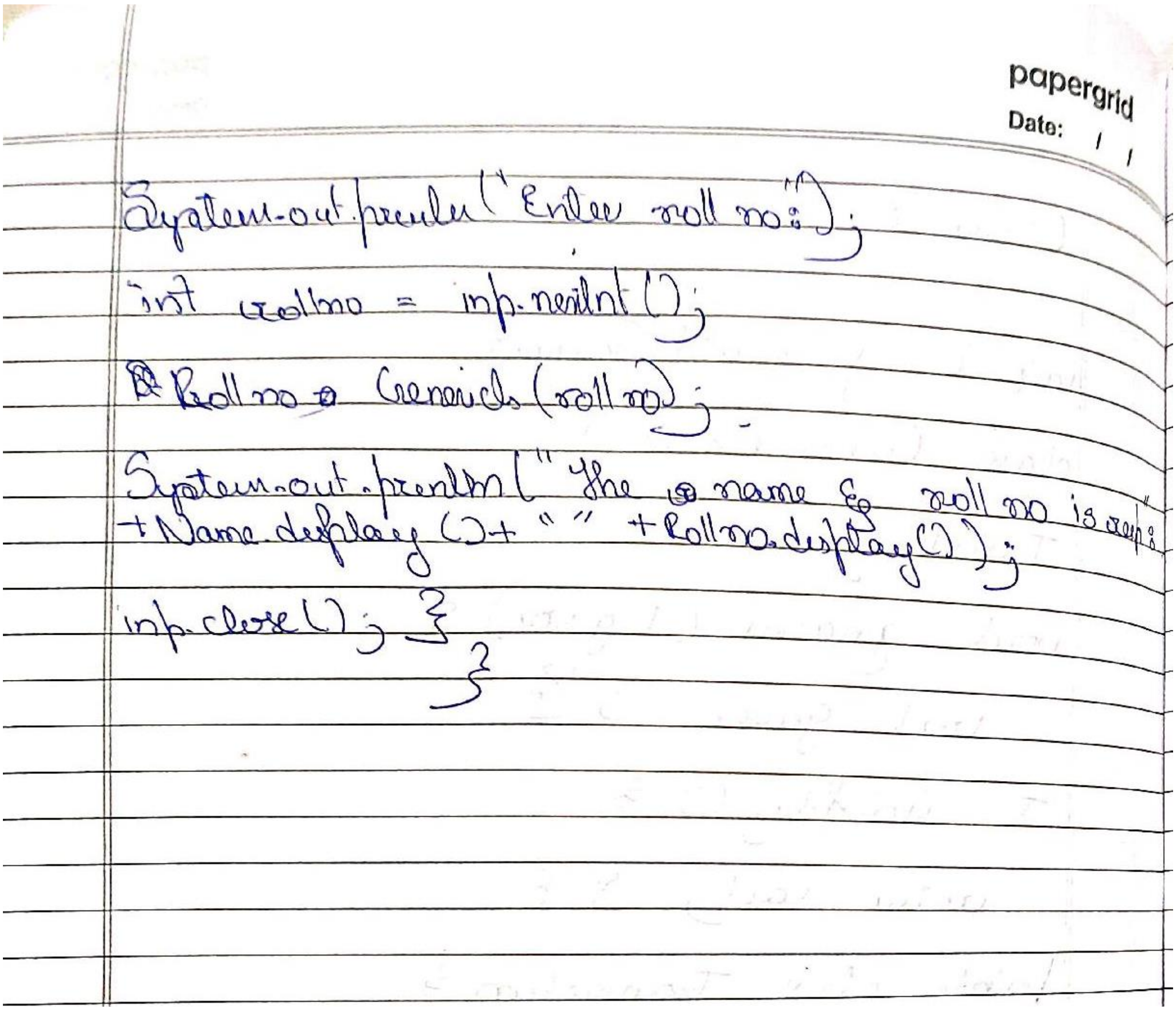


papergrid

Date: / /

Lab-7

```
import java.util.Scanner
class Generic <T> {
T val ;
void gvalue (T gval) {
    val = gval ; }
T Gdisplay () {
    return val; } }
public class Transaction {
public static void main (String args [])
Scanner inp = new Scanner(System.in);
Generics <Integer> Roll no = new Generic <Integer
Generics < String > String = new Generic<String> ();
System.out.println (" Enter name" );
 String sname = inp.nextline();
 Name Generics (sname) ;
```



papergrid

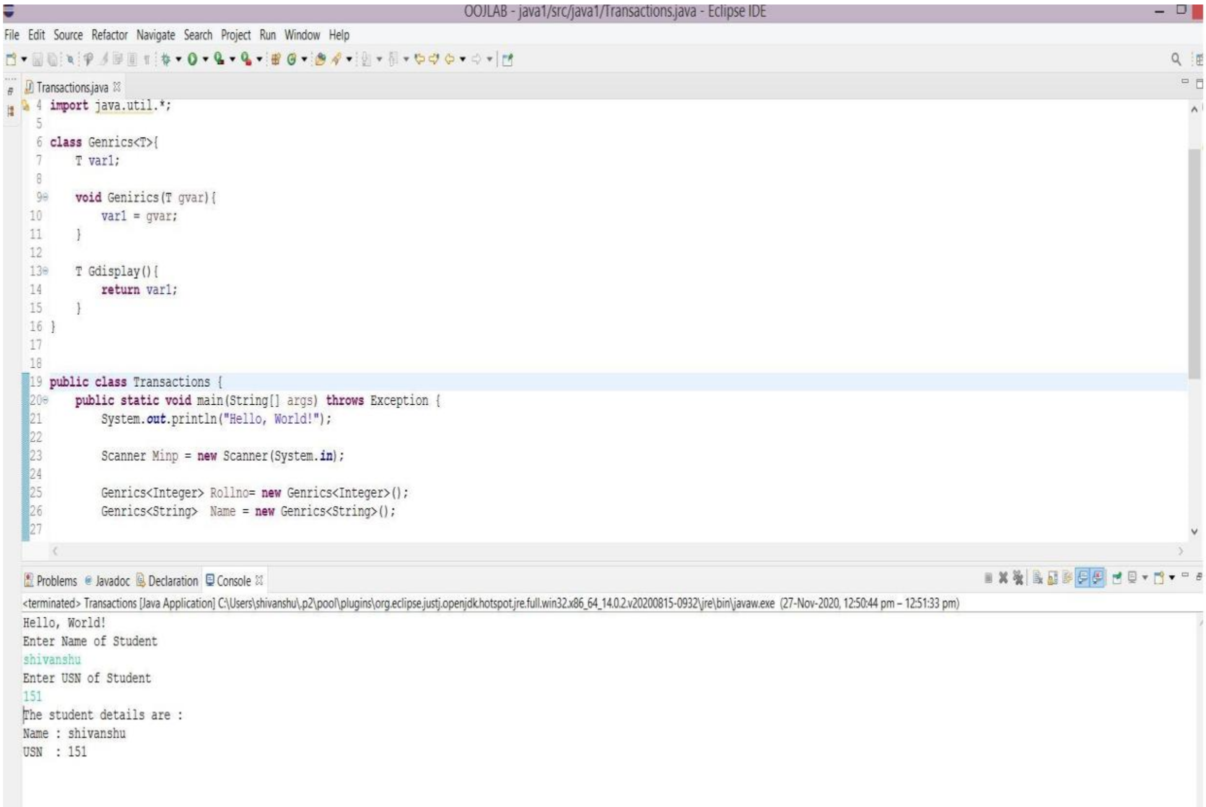
Date: / /

```
System.out.println("Enter roll no:");
int rollno = inp.nextInt();
Rollno = Generics(rollno);
System.out.println("The name & roll no is:"
+Name.display()+ " " +Rollno.display());
inp.close(); }
}
```

CODE:-

```
package labexam;
import java.util.Scanner;
class Mydata<T>
{ T i;
public void entered( T om)
{ om=i;
        }
public  T display(T i)
{
   return(i);
}
}

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

        System.out.println(" STRING INPUT");
        String sa=s.nextLine();
        System.out.println(" INT INPUT");
        int fa = s.nextInt();

        Mydata<Integer> m1= new Mydata<Integer>();
        Mydata<String> m2= new Mydata<String>();

        m1.entered(fa);
        m2.entered(sa);

        int out1 = m1.display(fa) ;
        String out2 = m2.display(sa);

        System.out.println("THE OUTPUT GOT USING GENERICS IS:" + out1 +" " +
out2);
        s.close();
    }
}
```

OUTPUT:-


OOJLAB - java1/src/java1/Transactions.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Transactions.java
Problems Javadoc Declaration Console

```
import java.util.*;

class Genrics<T>{
    T var1;

    void Genirics(T gvar){
        var1 = gvar;
    }

    T Gdisplay(){
        return var1;
    }
}


public class Transactions {
    public static void main(String[] args) throws Exception {
        System.out.println("Hello, World!");

        Scanner Minp = new Scanner(System.in);

        Genrics<Integer> Rollno= new Genrics<Integer>();
        Genrics<String>  Name = new Genrics<String>();

```

<terminated> Transactions [Java Application] C:\Users\shivanshu\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_14.0.2.v20200815-0932\jre\bin\javaw.exe (27-Nov-2020, 12:50:44 pm – 12:51:33 pm)

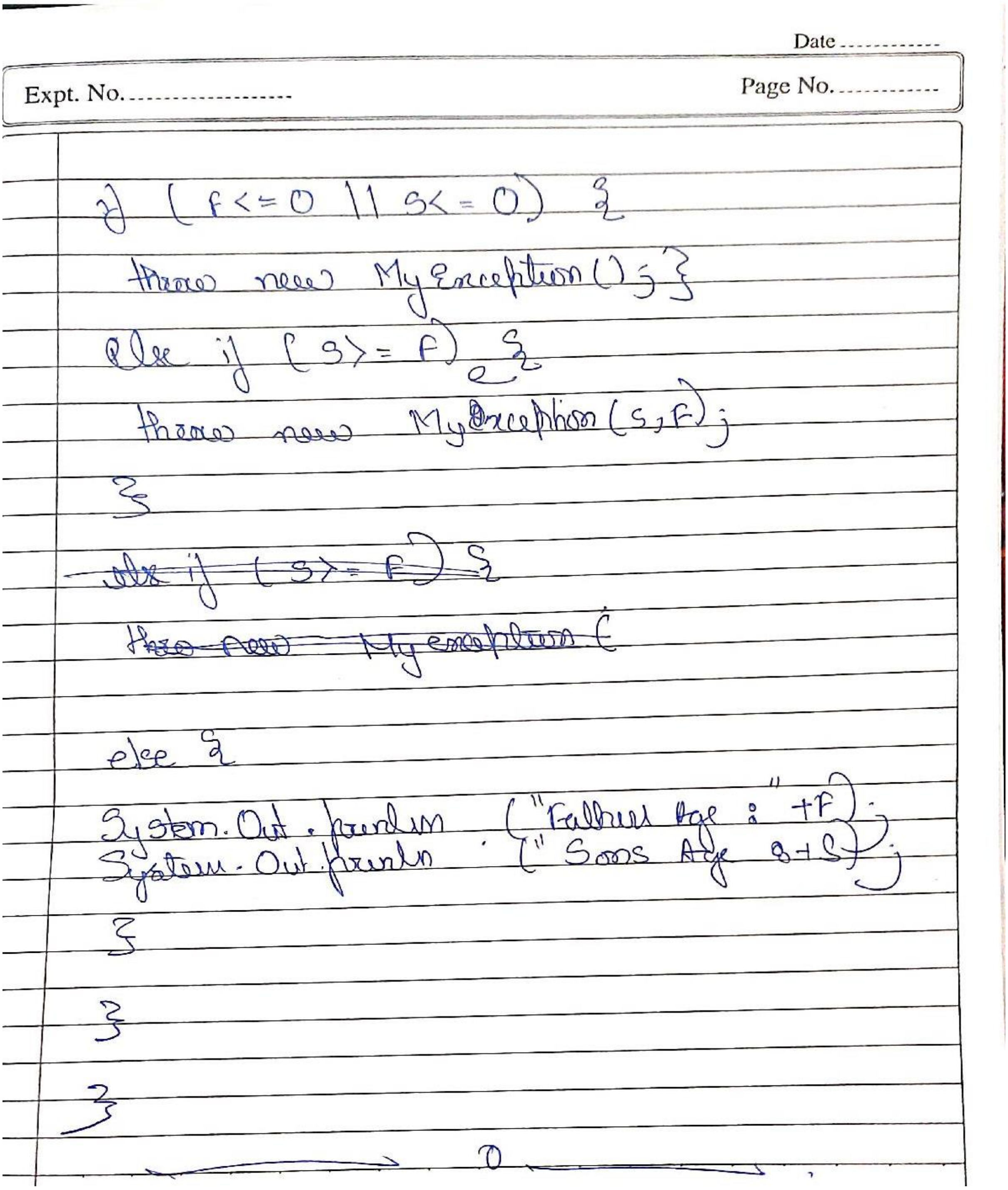
```
Hello, World!
Enter Name of Student
shivanshu
Enter USN of Student
151
The student details are :
Name : shivanshu
USN  : 151
```

LAB 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.

CIE Pakage (Student class) -

```
package cie ;

public class students {

    public int usn ;
    public string name;
    public int sem ;

    public students (int usn ; string name ; int sem)

    { this.usn = usn;
       this.name= name;
        this.sem= sem; }
// Constouct for init3//
    }

package cie ;

public class interenrals extendes Student {

    public int [ ] cie marks = new int [5] ;

    public internals (int usn, string name, int
       sem, int [ ] cie marks) {
       super( usn, name, sem);
       this.ciemarks = cie marks ; }
```

Teacher's Signature :



Date ....

Expt. No. .... Page No. ....

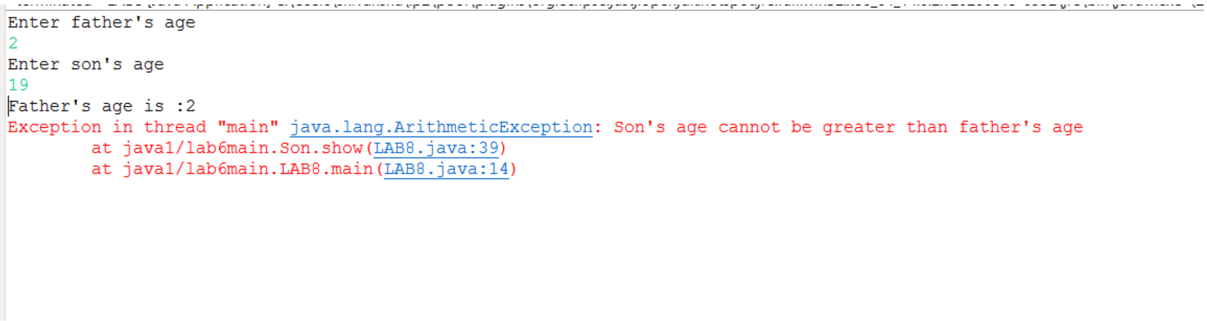
```
if ( f<=0 || s<= 0) {
  throw new MyException(); }
else if (s>= F) {
  throw new MyException(s,F);
}
~~else if ( s>= F) {~~
  ~~throw new Myexception (~~

else {
System.Out.println ("Fathers Age : " +F);
System.Out.println ("Sons Age "+S);
}
}
}
```

CODE;-
import
java.util.Scanner;

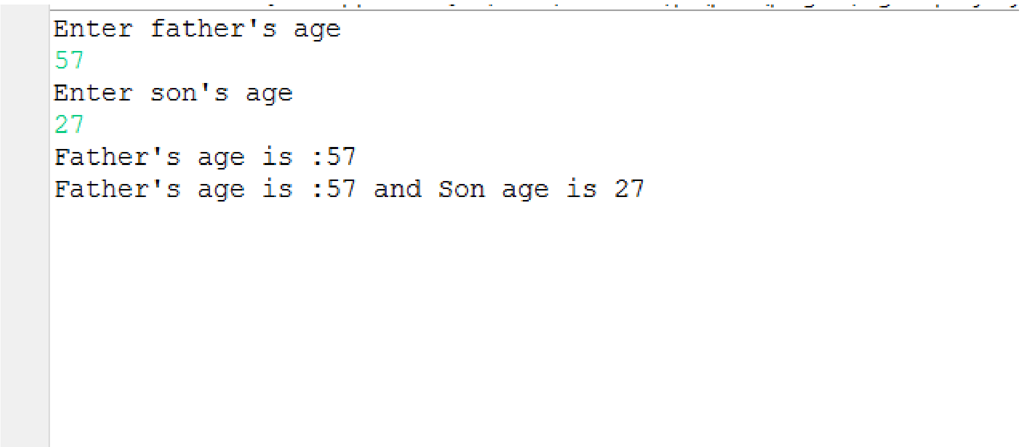
```
                        public class MyClass{
                            public static void main(String[] args){
                                Scanner sc=new Scanner(System.in);
                                System.out.println("Enter father's age");
                                int Fage=sc.nextInt();
                                System.out.println("Enter son's age");
                                int Sage=sc.nextInt();
```

```
Father o1=new Father(Fage);
Son o2=new Son(Sage,Fage);
o1.display();
o2.show();
    }
}
class Father{
    int age;
    Father(int age){
        this.age=age;
    }
    void display(){
    if(age<0)
        throw new ArithmeticException("wrong age");
    else
        System.out.println("Father's age is :"+age);
}
}
class Son extends Father{
    int SonAge;
    Son(int SonAge,int age){
        super(age);
        this.SonAge=SonAge;
    }
    void show(){
    if(SonAge>age )
        throw new ArithmeticException("Son's age cannot be greater
than father's age");
    else
    System.out.println("Father's age is :"+age+" and fathers age is
"+age);
}}
```

OUTPUT:-

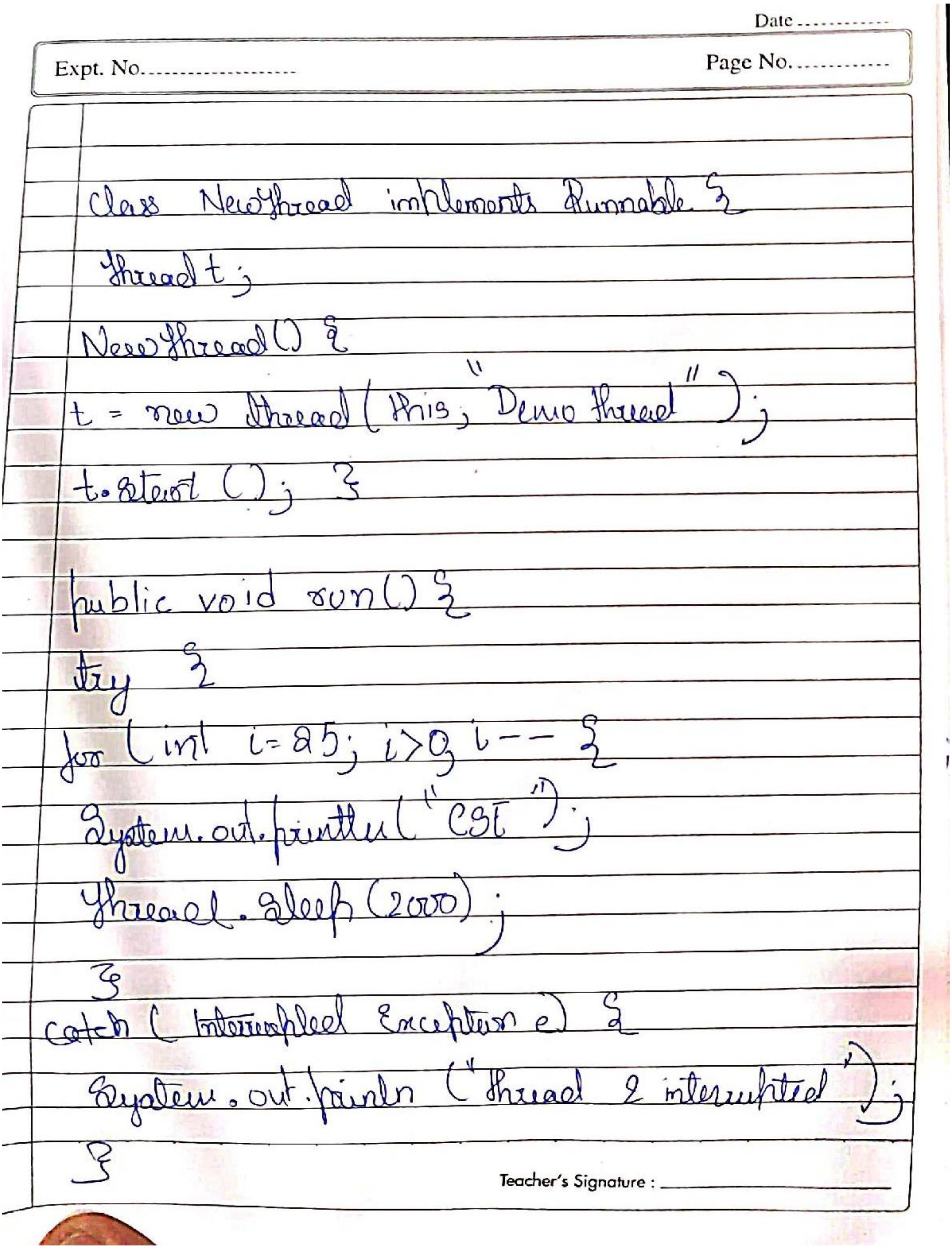


```
Enter father's age
2
Enter son's age
19
Father's age is :2
Exception in thread "main" java.lang.ArithmeticException: Son's age cannot be greater than father's age
	at java1/lab6main.Son.show(LAB8.java:39)
	at java1/lab6main.LAB8.main(LAB8.java:14)
```



```
Enter father's age
57
Enter son's age
27
Father's age is :57
Father's age is :57 and Son age is 27
```

LAB 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.

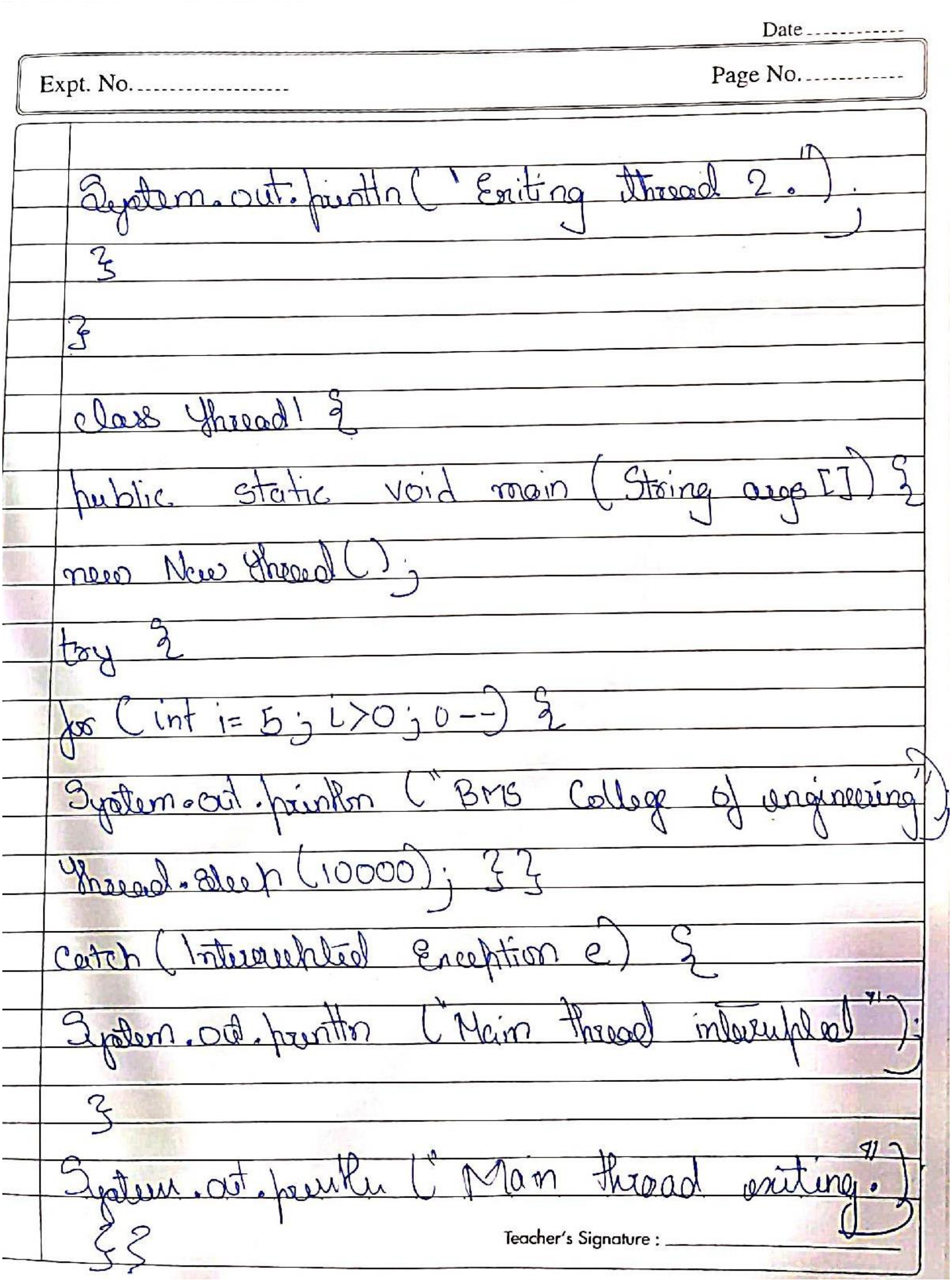


Date ....

Expt. No. .... Page No. ....

```
Class NewThread implements Runnable {
 Thread t;
NewThread() {
t = new Thread( this, "Demo thread" );
t.start (); }

public void run() {
try {
for (int i=25; i>0; i-- {
 System.out.println("CSE");
 Thread.sleep(2000);
 }
catch ( InterruptedException e) {
 System.out.println ("thread 2 interrupted");
 }
```

Teacher's Signature : ____________



Date

Expt. No.

Page No.

```
System.out.println("Exiting thread 2.");
 }
}
class Thread1 {
public static void main (String args[]) {
new New thread();
try {
for (int i= 5; i>0; i--) {
System.out.println ("BMS College of engineering");
Thread.sleep (10000); }}
catch (Interrupted Exception e) {
System.out.println ("Main thread interupted");
 }
System.out.println ("Main thread exiting.");
 }}
```

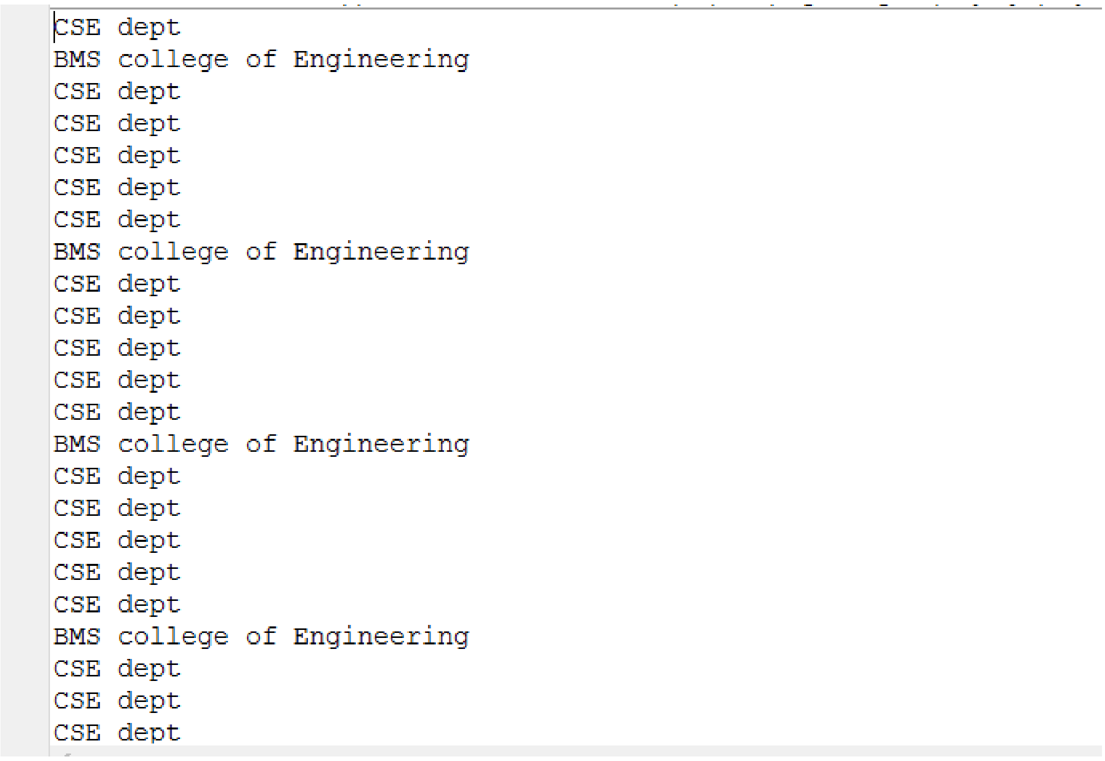
Teacher's Signature :

CODE:-

```
class Thread1 implements Runnable{
    Thread t;
    String name;
```

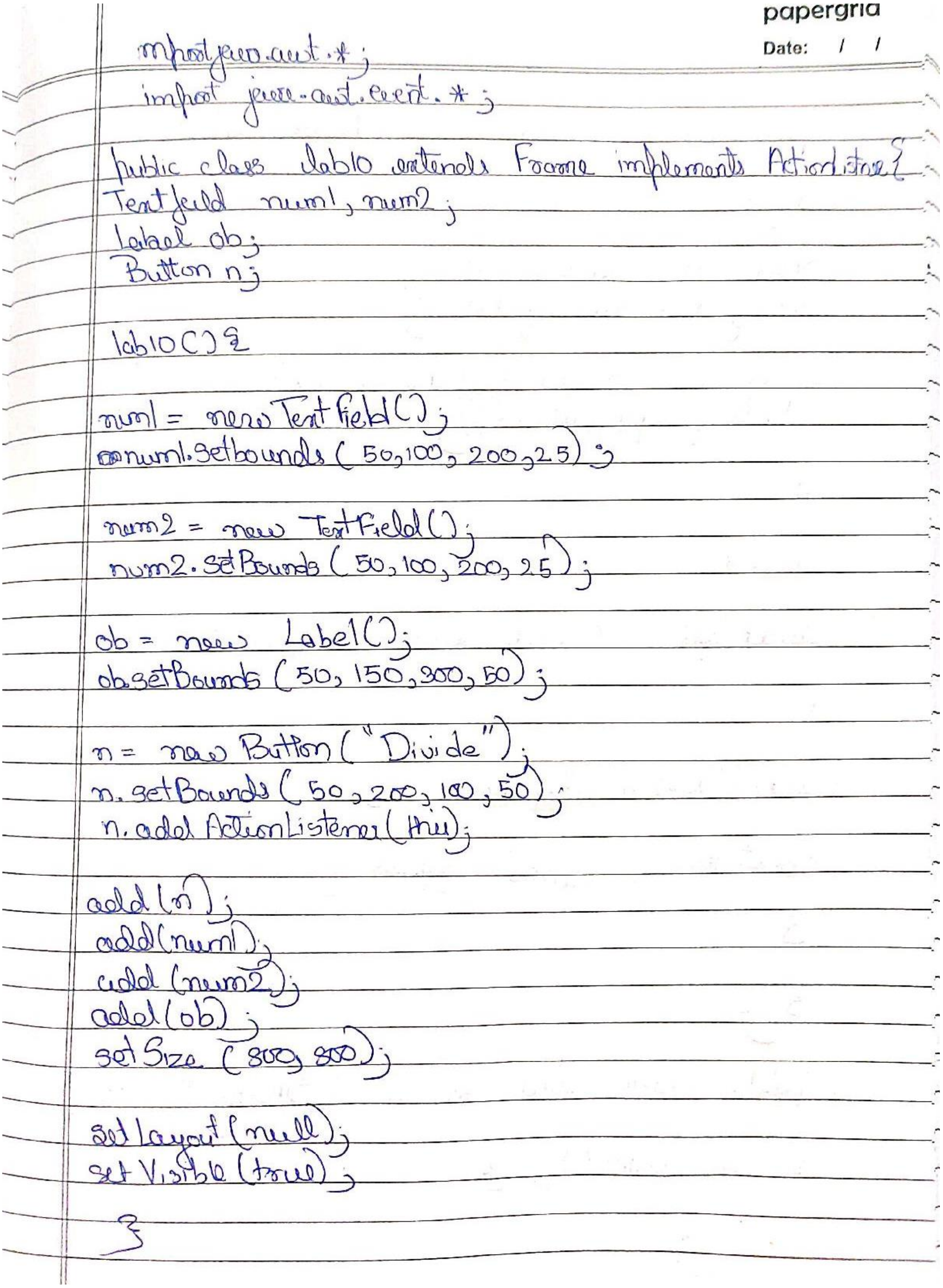
```
   Thread1(String name){
      this.name = name;
      t = new Thread(this,this.name);
      t.start();
   }
   public void run(){
      try{
         for(int i=0;i<20;i++){
            System.out.println("CSE dept");
            Thread.sleep(2000);
         }
      }catch(InterruptedException e){
         System.out.println(e);
      }
   }
}

class Thread2 implements Runnable{
   Thread t;
   String name;
   Thread2(String name){
      this.name = name;
      t = new Thread(this,this.name);
      t.start();
   }
   public void run(){
      try{
         for(int i=0;i<5;i++){
            System.out.println("BMS college of Engineering");
            Thread.sleep(10000);
         }
      }catch(InterruptedException e){
         System.out.println(e);
      }
   }
}

class LAB8{
   public static void main(String[] args){
      Thread1 obj1 = new Thread1("Dept. name");
      Thread2 obj2 = new Thread2("College name");
      //System.out.println(obj1.name+" "+obj1.t.isAlive());
      //System.out.println(obj2.name+" "+obj2.t.isAlive());
      try{
         obj1.t.join();
         obj2.t.join();
      }catch(Exception e){
         System.out.println("Interrupted");
      }
   }}
```

OUTPUT:-

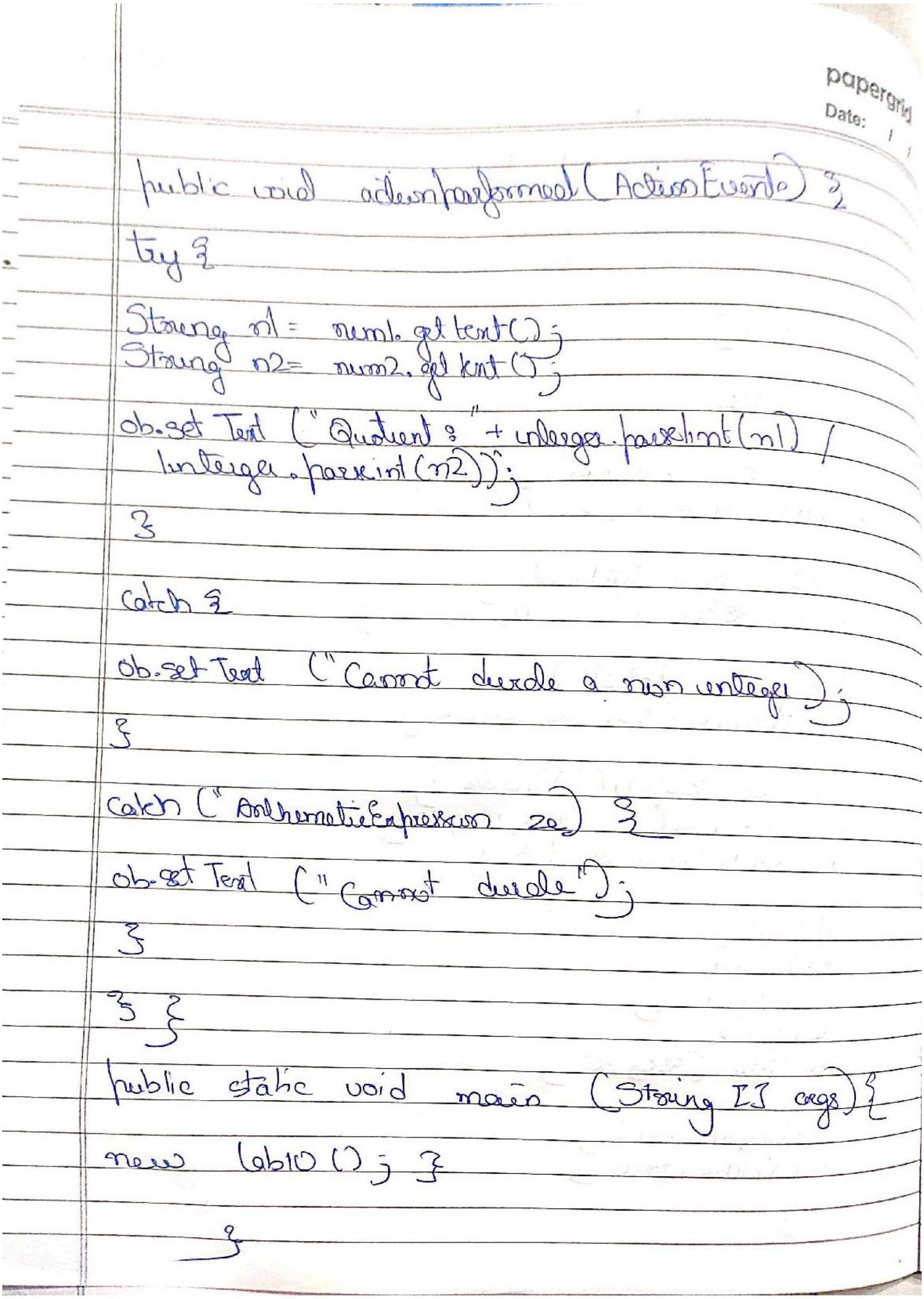


```
CSE dept
BMS college of Engineering
CSE dept
CSE dept
CSE dept
CSE dept
CSE dept
BMS college of Engineering
CSE dept
CSE dept
CSE dept
CSE dept
CSE dept
BMS college of Engineering
CSE dept
CSE dept
CSE dept
CSE dept
CSE dept
BMS college of Engineering
CSE dept
CSE dept
CSE dept
```

LAB10:- 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.



```
mport java.awt.*;
import java.awt.event.*;

public class lab10 extends Frame implements ActionListener{
TextField num1, num2;
Label ob;
Button n;

lab10(){

num1 = new TextField();
num1.setbounds(50,100, 200,25);

num2 = new TextField();
num2.setBounds(50,100,200,25);

ob = new Label();
ob.setBounds(50,150,300,50);

n = new Button("Divide");
n.setBounds(50,200,100,50);
n.add ActionListener(this);

add(n);
add(num1);
add(num2);
add(ob);
setSize(800,800);

setLayout(null);
setVisible(true);
}
```



papergrid

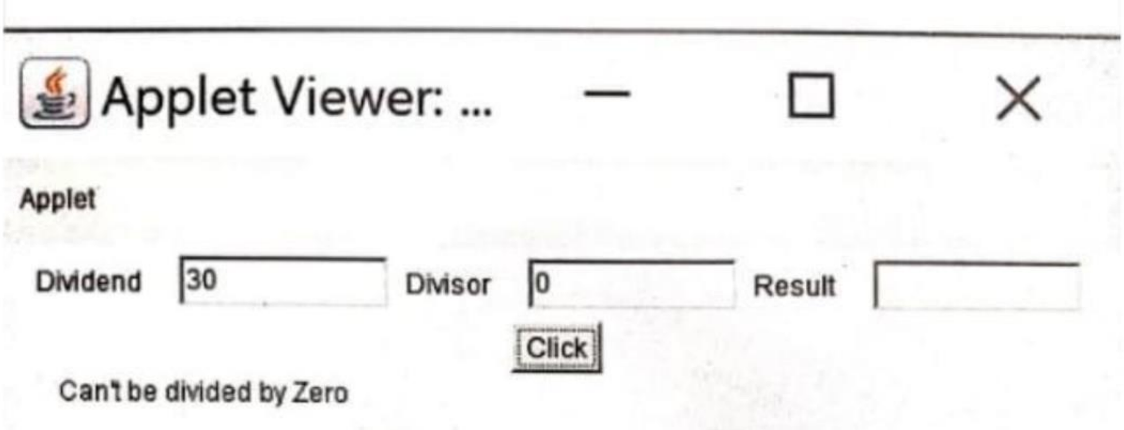
Date: / /

```
public void actionPerformed(ActionEvent e) {
try {
String n1 = num1.getText();
String n2 = num2.getText();
ob.setText ("Quotient : " + Integer.parseInt(n1) /
  Integer.parseInt(n2));
}
catch {
ob.setText ("Cannot divide a non integer );
}
catch (" ArithemeticExpression ze) {
ob.setText ("Cannot divide");
}
} }
public static void main (String [] args){
new lab10 (); }
   }
```

CODE:-
import

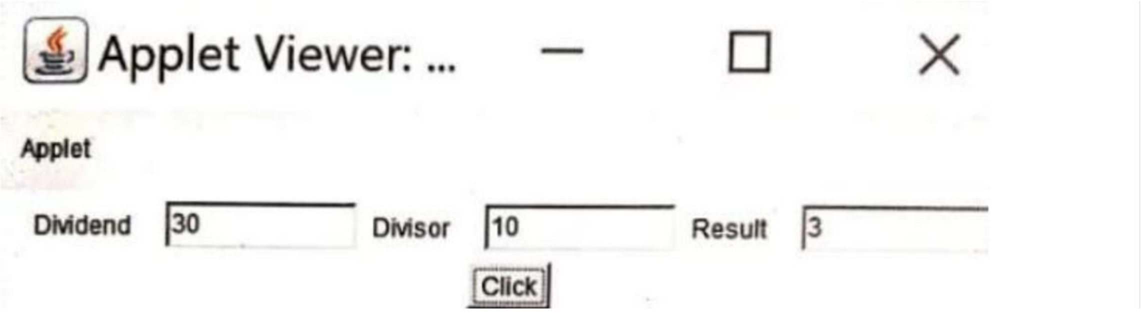
```
java.awt.*;
        import java.awt.event.*;
        import java.applet.*;
        /*<applet code="DivisionExample"width=230 height=250></applet>*/
         class DivisionExample extends Applet implements ActionListener {
            String msg;
            TextField num1, num2, res;
            Label l1, l2, l3;
            Button div;
            public void init() {
                l1 = new Label("Dividend");
                l2 = new Label("Divisor");
                l3 = new Label("Result");
                num1 = new TextField(10);
                num2 = new TextField(10);
                res = new TextField(10);
                div = new Button("Click");
                div.addActionListener(this);
                add(l1);
                add(num1);
                add(l2);
                add(num2);
                add(l3);
                add(res);
                add(div);
            }
            public void actionPerformed(ActionEvent ae) {
                String arg = ae.getActionCommand();
                int num1 = 0, num2 = 0;
                if (arg.equals("Click")) {
                    if (this.num1.getText().isEmpty() &&
        this.num2.getText().isEmpty()) {
                        msg = "Enter the valid numbers!";
                        repaint();
                    } else {
                        try {
                            num1 =
        Integer.parseInt(this.num1.getText());
                            num2 =
        Integer.parseInt(this.num2.getText());
                            int num3 = num1 / num2;
                            res.setText(String.valueOf(num3));
                            msg = "Operation Succesfull!!!";
                            repaint();
                        } catch (NumberFormatException ex) {
                            System.out.println(ex);
                    res.setText("");
                    msg = "NumberFormatException -
Non-numeric";

                    repaint();
                } catch (ArithmeticException e) {
                    System.out.println("Can't be divided
by Zero" + e);

                    res.setText("");
                    msg = "Can't be divided by Zero";
                    repaint();
                }
            }
        }
    }
public  void paint(Graphics g) {
        g.drawString(msg, 30, 70);
    }

}
```

OUTPUT:-



Dividend 30 Divisor 0 Result

Can't be divided by Zero



Dividend 30 Divisor 10 Result 3