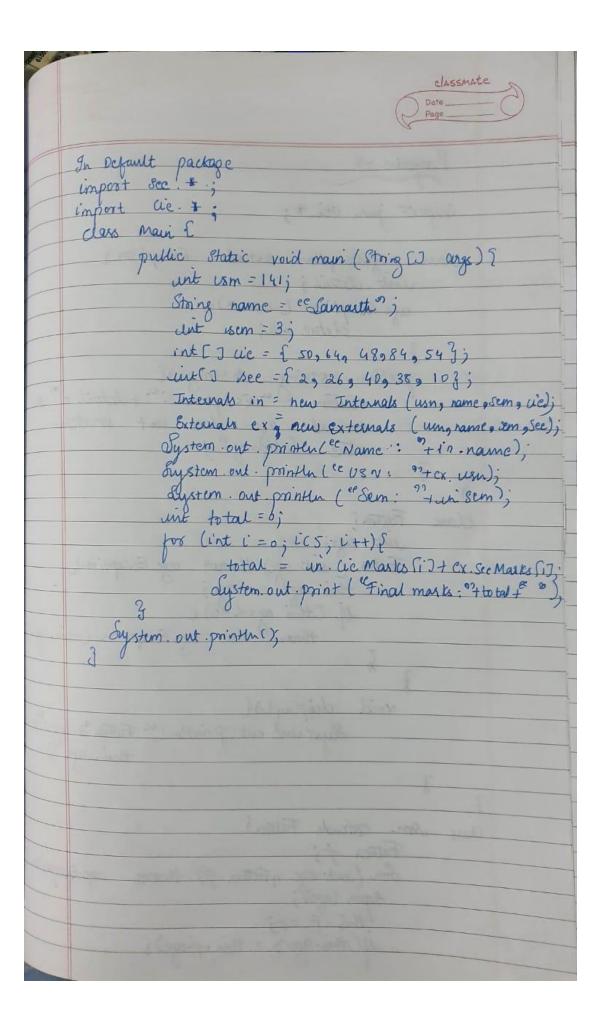
Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External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

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

```
import CIEp.*;
public class Externals extends Student{
  public int[] seeMarks = new int[5];
  public Externals(int usn,String name,int sem,int[] seeMarks){
    super(usn,name,sem);
    this.seeMarks = seeMarks;
  }
}
import CIEp.*;
import SEEp.*;
import java.util.*;
public class Main7{
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    Externals[] e = new Externals[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 Externals(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].seeMarks[j]+in[i].cieMarks[j];
    System.out.print("Final marks: "+total+" ");
}
System.out.println();
}</pre>
```

(AB Program - 6 (IE parkage (Student class) package cie; Public class Student & public unt usn; public String name; public unt sen; public Student Cunt usno string into this usn = usn; this name = name; this sem = sem, 1/ internals class package lie; public class unternals extends Student ? public int[] ine marks = new int[s]; public internals (ent using Storing name, int 8em q int] are marks) & Supres (Usn, name, sem); this . cie marks = lie marks; SEE padcage (external clays) parkage usee; umposta cie. *; public Class Externals extends Studentic int [] See Masks - new int[]; Public externals (Int using String name , Int int[] sec marks) { Super (using name , wome this. Sec Marks = See Marks)



```
C:\Users\Samarth\Documents\pac2>java Main7.java

C:\Users\Samarth\Documents\pac2>java Main7
Enter the Number of students : 10
Enter the details of the student 1:
Enter usn of the student : SAM
Enter semester of the student : 3
Enter he 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 of the course 3: 14
Enter the SEE marks of the course 3: 67
Enter the SEE marks of the course 3: 67
Enter the SEE marks of the student 2:
Enter the SEE marks of the course 3: 67
Enter the SEE marks of the student 2:
Enter semester of the student 2:
Enter usn of the student : 3
Enter semester of the student : 3
Enter marks of the course 1: 15
Enter marks of the course 2: 20
Enter marks of the course 2: 20
Enter marks of the course 3: 38
Enter the SEE marks of the course 2: 78
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.

```
genrics
import java.util.*;

class Genrics<T>{
    T var1;

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

    T Gdisplay(){
       return var1;
    }
}
```

```
public class App {
  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>();
    System.out.println("Enter Name of Student");
    String Sname = Minp.nextLine();
    Name.Genirics(Sname);
    System.out.println("Enter USN of Student");
    int Sno = Minp.nextInt();
    Rollno.Genirics(Sno);
    System.out.println("The student details are :");
    System.out.println("Name : "+ Name.Gdisplay());
    System.out.println("USN : "+ Rollno.Gdisplay());
    Minp.close();
  }
}
```

Program 7: import java . util. #; class Gentics (T > { T vail; void Geninies (T gran) { Val = gran; T a display () { return var1; public class App & public Static void main (String [] args) throws Exception (System. out. printle (ce pello, world!"); ofcanner Minp = new 8 canner Lystem in Grennics (Integer) Rollino = new Gernics (Inte Gennics (String > Name = new Gennis (String System out printle Lee Ender name of Student 8 tring Sname = Minp. nextline (); 8 Name. Germinics (Sname); dystom. out. println (ex Enter USN of Student int Sno = Minp. next Int (); Rollno Creamico (Sna); System out printly lec The student details me System out printly Lec Name: 97 + Name Indispla Sufferent out grinth (" USN: ">+ polling bidisply (Minp. doset);

```
Microsoft Windows [Version 10.0.19041.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Samarth>cd C:\Users\Samarth\Documents

C:\Users\Samarth\Documents>javac App.java

C:\Users\Samarth\Documents>java App
STRING INPUT
Praveen
INT INPUT
123
THE OUTPUT GOT USING GENERICS IS:123 Praveen

C:\Users\Samarth\Documents>exit_
```

//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 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.

```
import java.util.*;

class ageException extends Exception{
  int detail;
  ageException(int a){
    detail = a;
  }

  public String toString(){
    return "Exception:"+detail+" the enterred age does not match the category";
  }
```

```
}
class Father{
  int age;
  Father(int age) throws ageException{
    this.age = age;
    if(this.age<=0){
      throw new ageException(this.age);
    }
  }
  void display(){
    System.out.println("Father's age:"+this.age);
  }
}
class Son extends Father{
  Father f;
  Son(int age,Father f) throws ageException{
    super(age);
    this.f = f;
    if(this.age>=this.f.age){
      //System.out.println(f.age);
      throw new ageException(this.age);
    }
  }
  void display(){
    this.f.display();
    System.out.println("Son's age:"+this.age);
  }
}
public class Lab{
```

```
public static void main(String[] args){
    try{
        Scanner input = new Scanner(System.in);
        Father f = new Father(input.nextInt());
        Son s = new Son(input.nextInt(),f);
        s.display();
    }catch(Exception e){
        System.out.println(e);
    }
}
```

Prograin -8 impost java. Util. *; class age Exception extends Exception & unt detail; age Exception (unt a) { detal = a; public stoine to stoine () {

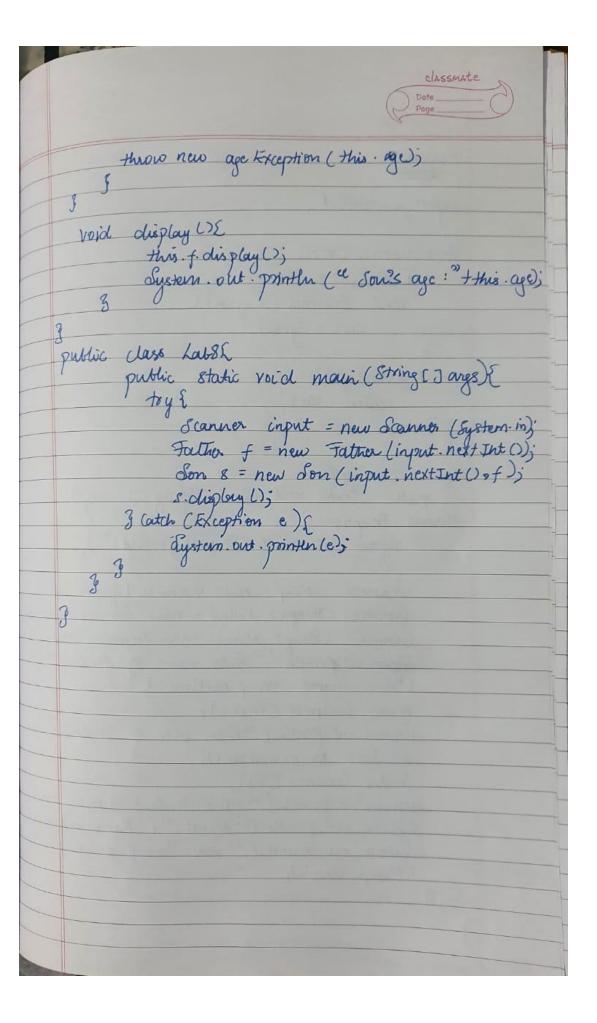
return entered are does not match

the Cattgory"; class Father ? Faltre (unt age) throws age Exceptions

this age = age;

If (this age <=0) {

throw new age Except on (this ut age; void display () & Rystem out Printler (ec Father's the age) Class son extends Father ? Son (int age grotther f) throws age kx g super (age); this f = f; if Cthis age > = this f-age) &

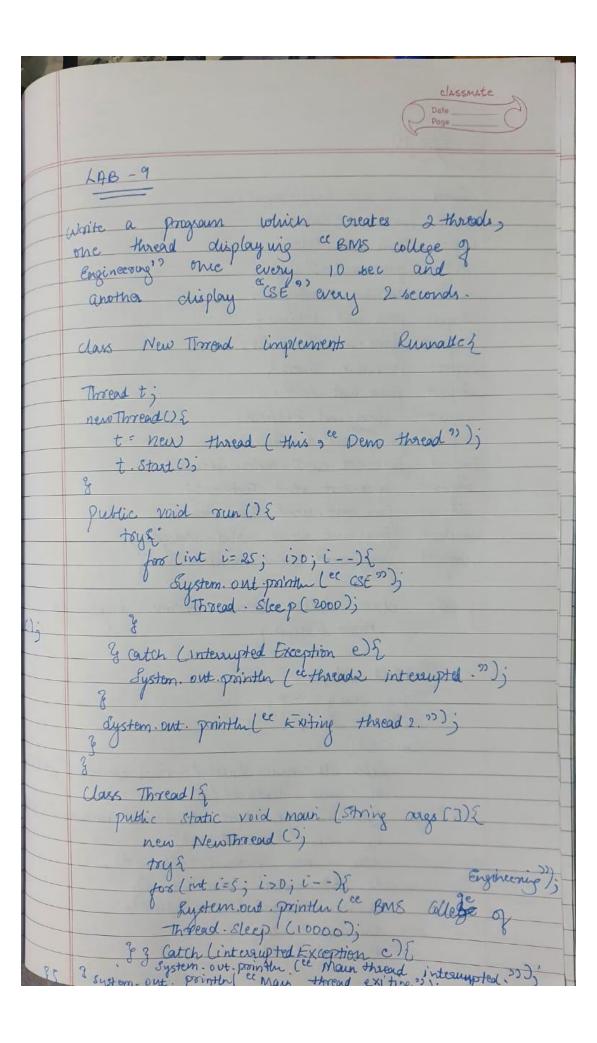


```
Microsoft Windows [Version 10.0.19041.685]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\Samarth>cd C:\Users\Samarth\Documents
C:\Users\Samarth\Documents>javac Lab.java
C:\Users\Samarth\Documents>javac Lab.javac Lab
```

// LAB 9

```
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("thread2 interrupted.");
}
System.out.println("Exiting thread2.");
}
}
class Thread1 {
```

```
public static void main(String args[]) {
  new NewThread();
  try {
  for(int i = 5; i > 0; i--) {
    System.out.println("BMS College of Engineering");
    Thread.sleep(10000);
  } } catch (InterruptedException e) {
    System.out.println("Main thread interrupted.");
  }
  System.out.println("Main thread exiting.");
}
```



```
## Command Homoga (Version 18.0.19041.638)

(c) 2828 Microsoft Kindows (Version 18.0.19041.638)

(c) 2828 Microsoft Corporation. All rights reserved.

C:\Users\Samarth\Documents\java\Samarth\Documents

C:\Users\Samarth\Documents\java\Samarth\Documents

C:\Users\Samarth\Documents\java\Samarth\Documents\java\Samarth\Documents\java\Samarth\Documents\java\Samarth\Documents\java\Samarth\Documents\java\Samarth\Documents\java\Samarth\Documents\java\Samarth\Documents\java\Samarth\Documents\java\Samarth\Documents\java\Samarth\Documents\java\Samarth\Documents\java\Samarth\Documents\java\Samarth\Samarth\Documents\java\Samarth\Samarth\Samarth\Documents\java\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth\Samarth
```

//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.

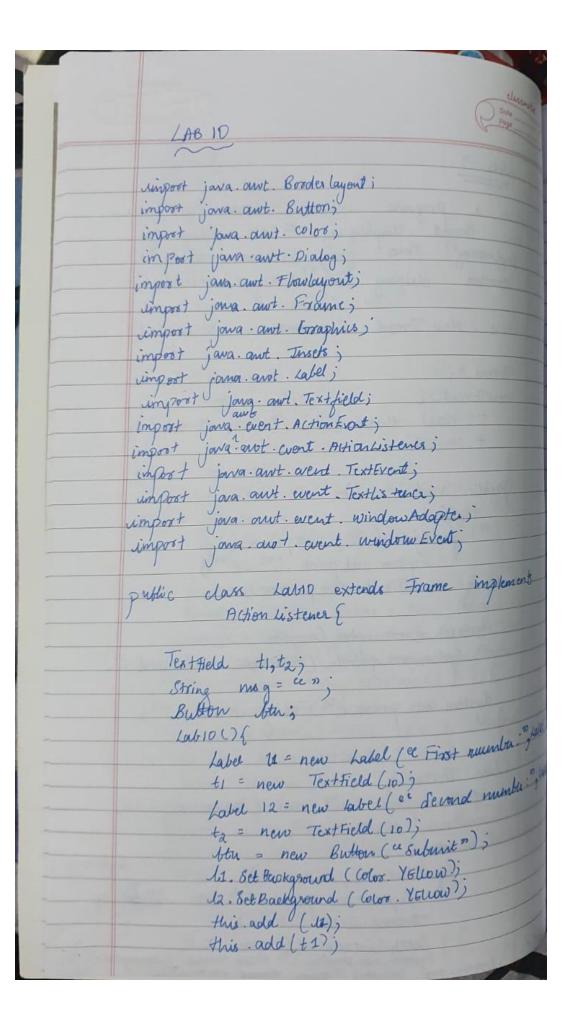
```
import java.awt.BorderLayout;
import java.awt.Button;
import java.awt.Color;
import java.awt.Dialog;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Graphics;
import java.awt.Insets;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.TextEvent;
import java.awt.event.TextListener;
```

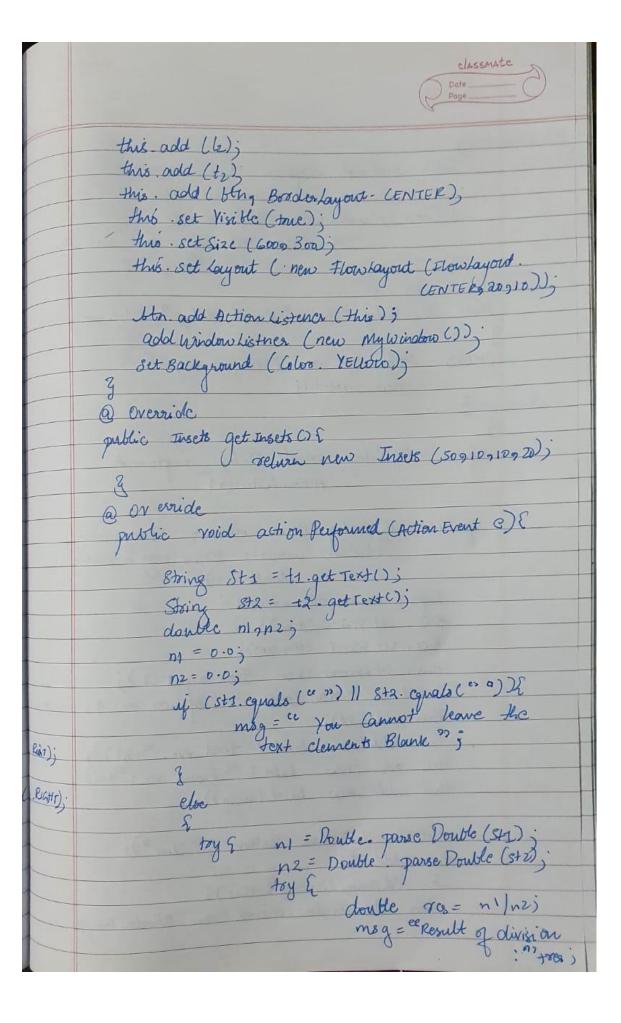
```
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
public class Lab10 extends Frame implements ActionListener{
       TextField t1,t2;
       String msg="";
       Button btn;
       Lab10(){
               Label I1 = new Label("First Number: ",Label.RIGHT);
               t1 = new TextField(10);
               Label I2 = new Label("Second Number: ",Label.RIGHT);
               t2 = new TextField(10);
               btn = new Button("Submit");
               //Label I = new Label("Updates:");
               l1.setBackground(Color.YELLOW);
               12.setBackground(Color.YELLOW);
               //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.YELLOW);
       //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 I1 = new Label("
                                           Updates on the result:
                                                                       ");
                //l1.setSize(300, 20);
                this.add(I1);
                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);
       }
}
```





Carch (Arithematic Exception e1) [mag = cl. tosting (); 3 Cotch (Number Format Exception cr) ? msg = a Enter they not and not things " New My Dialog Cthis , a Result Dialog , take my public static void manin (String [] args) { new Lauto (); Class My Dialog extends Dialog implements Action Listener [public My Dialog CFrame owner of Storing title, booken models string may a double double 22 2 Super (numers title of modal) this. Set Visible (true) this, set Size (300, 400)) this. Set Layout (new Flow Layout ()) Label b = new label (ec updates on he the this add (11); this add (new tatel (exfirst No: "+11))
this add (new Label Cee Second no: "+m) this add (new Label (msg)); button b = new Button (ce close "); this add (b)) b. add Action Listener (this) ; this add Window Listener (new Window Adapto

