

## **Week 9-Lab Program 6**

**Solve this program and write the procedure you have used to execute this in your observation**

**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 cie;

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 cie;

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 see;  
  
import cie.*;  
  
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 cie.*;  
import see.*;  
import java.util.*;  
class Main{  
    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++){
    System.out.println("Enter Usn,Name,Sem");
    int usn1 = input.nextInt();
    String name1 = input.nextLine();
    int sem1 = input.nextInt();
    int[][] cie = new int[2][5];
    int[][] see = new int[2][5];
    System.out.println("Enter Cie Marks in 5 Subjects");

    for(int j=0;j<5;j++){
        cie[i][j] = input.nextInt();
    }

    System.out.println("Enter See Marks in 5 subjects");

    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+"\nUSN: "+e[i].usn+"\nsem: "+e[i].sem);

    for(int j=0;j<5;j++){
        total = (e[i].seeMarks[j]/2)+in[i].cieMarks[j];
        System.out.print("Final marks: "+total+" ");
    }
}

```

```
    }  
  
    System.out.println();  
  
}  
  
}  
  
}
```

```
C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 6>java Main  
Enter Usn,Name,Sem  
1471  
Shashank  
3  
Enter Cie Marks in 5 Subjects  
50  
49  
45  
46  
47  
Enter See Marks in 5 subjects  
100  
98  
89  
78  
90  
Name: Shashank  
USN: 1471  
sem: 3  
Final marks: 100 Final marks: 98 Final marks: 89 Final marks: 85 Final marks: 92  
  
Enter Usn,Name,Sem  
1472  
Sharma  
3  
Enter Cie Marks in 5 Subjects  
49  
48  
47  
40  
50  
Enter See Marks in 5 subjects  
100  
99  
87  
100  
89  
Name: Sharma  
USN: 1472  
sem: 3  
Final marks: 99 Final marks: 97 Final marks: 90 Final marks: 90 Final marks: 94
```

## CAB Prog-6

### CIE Package (student)

```

package cie;
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;
    }
}

```

### (Internals)

```

package cie;
public class Internals extends Student {
    public int [3] clemarks = new int [5];
    public Internals (int usn, String name, int sem,
                     int [3] clemarks) {
        super (usn, name, sem);
        this.clemarks = clemarks;
    }
}

```

(See Package)

package See;

import cie.\*;

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;

3

3

(Main program )

import cie.\*;

import spp.\*;

import java.util.\*;

class marks

public static void main (String [] args) {

Scanner input = new Scanner (System.in);

externals [] e = new externals [2];

internals [] i = new internals [2];

for (int i=0; i<2; i++) {

System.out.println ("Enter USN, Name, Sem");

Seqno);

int usn = input.nextInt();

String name = input.nextLine();

int sem = input.nextInt();

int [] [] i [] = new int [2] [5];

int [] [] e [] = new int [2] [5];

System.out.println ("Enter 5 subjects ");

Subject ");



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

c[i][j] = input.nextInt();

}

System.out.println("Enter see marks in 5 subjects");

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

see[i][j] = input.nextInt();

}

e[i] = new externals (usn, name, sem, see[i]);

i[n][i] = new internals (usn, name, sem, c[i][j]);  
int total=0;

System.out.println("Name: " + e[i].name + " | USN: " +  
e[i].usn + " | SEM: " + e[i].sem);

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

total = (e[i].seemarks[i]/2) + i[n][i].cemarks[i];

System.out.println("Final Marks : " + total + " ");

}

System.out.println();

}

3

3

## Program 7

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

```
class Gen<T,S>{

    private T obj;

    private S obj1;

    Gen(T value,S value2){

        obj = value;

        obj1 = value2;

    }

    T getObj(){

        return obj;

    }

    S getObj1(){

        return obj1;

    }

    void objType(){

        System.out.println("The type of object "+obj.getClass().getName());

    }

    void objType1(){

        System.out.println("The type of object "+obj1.getClass().getName());

    }

}

class Main{

    public static void main(String[] args){

        Gen<Integer,Double> ob = new Gen<Integer,Double>(88,88.889);
```

```
    ob.objType();
    System.out.println("Object Value "+ob.getObj());
    ob.objType1();
    System.out.println("Object Value "+ob.getObj1());
    Gen<String,Integer> ob2 = new Gen<String,Integer>("SHashank",12);
    ob2.objType();
    System.out.println("Object Value "+ob2.getObj());
    ob2.objType1();
    System.out.println("Object Value "+ob2.getObj1());
}

}
```

```
C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 7>javac Main.java
C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 7>java Main
The type of object java.lang.Integer
Object Value 88
The type of object java.lang.Double
Object Value 88.889
The type of object java.lang.String
Object Value SHashank
The type of object java.lang.Integer
Object Value 12
```

## CAB PROG-7

Class Gen < T, S > {

    private T obj;

    private S obj1;

    Gen (T value, S value2) {

        obj = value;

        obj1 = value2;

}

    T getobj () {

        return obj;

}

    T getobj1 () {

        return obj1;

}

    void objType () {

        System.out.println("Type of object " + obj.getClass().getName());

}

    void objType1 () {

        System.out.println("Type of object " +

            obj1.getClass().getName());

}

}

Class Main

public static void main (String [] args) {

    Gen < Integer, Double > ob = new Gen

< Integer, Double >

(88, 88.889)

    ob. objType();

System.out.println("object value " + ob.getobj());

ob.objType();

System.out.println("object value " + ob.getobj());

Gen<String, Integer> ob = new Gen<String, Integer>("Shashank", 12);

ob.objType();

System.out.println("object value " + ob.getobj());

ob.objType();

System.out.println("object value " + ob.getobj());

3

3

Java

Variables

1

2

3

4

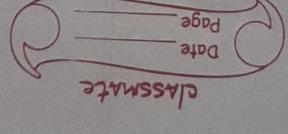
5

6

7

8

9



## Program 8

2. 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 entered age has a problem";  
  
    }  
  
}  
  
class Father{  
  
    int age;  
  
    Father(int age) throws ageException{  
  
        this.age = 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;
}

void display(){
    this.f.display();
    System.out.println("Son's age:"+this.age);
}

}

public class Main{

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

        System.out.println("Enter Father Age");
        Father f = new Father(input.nextInt());

        System.out.println("Enter Son Age");
        Son s = new Son(input.nextInt(),f);

        try{
            if(s.age>=f.age)
                throw new ageException(s.age);
            if(f.age<=0)
                throw new ageException(f.age);
            if(s.age<=0)
                throw new ageException(s.age);
            s.display();
        }catch(Exception e){
    }
}
```

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

    }

}

}

C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 8>javac Main.java
C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 8>java Main
Enter Father Age
-1
Enter Son Age
-3
Exception :-1 The entered age has a problem
C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 8>java Main
Enter Father Age
18
Enter Son Age
15
Father's age:18
Son's age:15

C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 8>java Main
Enter Father Age
20
Enter Son Age
30
Exception :30 The entered age has a problem
C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 8>
```

## LAB PROG-8

```
import java.util.*;
```

```
class ageException extends Exception {
```

```
    int detail;
```

```
    ageException (int a) {
```

```
        detail = a;
```

```
}
```

```
    public String toString () {
```

```
        return "Exception" + detail + "Entered age
```

has a problem"

```
}
```

```
}
```

```
class father {
```

```
    int age;
```

```
    father (int age) throws ageException {
```

```
        this.age = 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;
```

```
}
```

```
void display () {
```

```
    this.f.display ();
```

```
    System.out.println ("Son's age" + this.age);
```

3

public class Maths

public static void main (String [ ] args) throws  
ageException

8

Scanner input = new Scanner (System.in);  
System.out.println ("Enter father age");  
Father f = new Father (input.nextInt());  
System.out.println ("Enter son age");  
Son s = new Son (input.nextInt());

try {

if (~~(~~ s.age >= f.age)

throw new ageException (s.age);

if (f.age <= 0)

throw new ageException (f.age);

if (s.age <= 0)

throw new ageException (s.age);

s.display();

3 catch (Exception e) {

System.out.println (e);

3

3

3

**LAB PROGRAM:**

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 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");
```

```
                Thread.sleep(2000);
```

```
}
```

```
}catch(InterruptedException e){
```

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

```
}
```

```
}
```

```
class Thread2 implements Runnable{
```

```
    Thread t;
```

```
    String name;
```

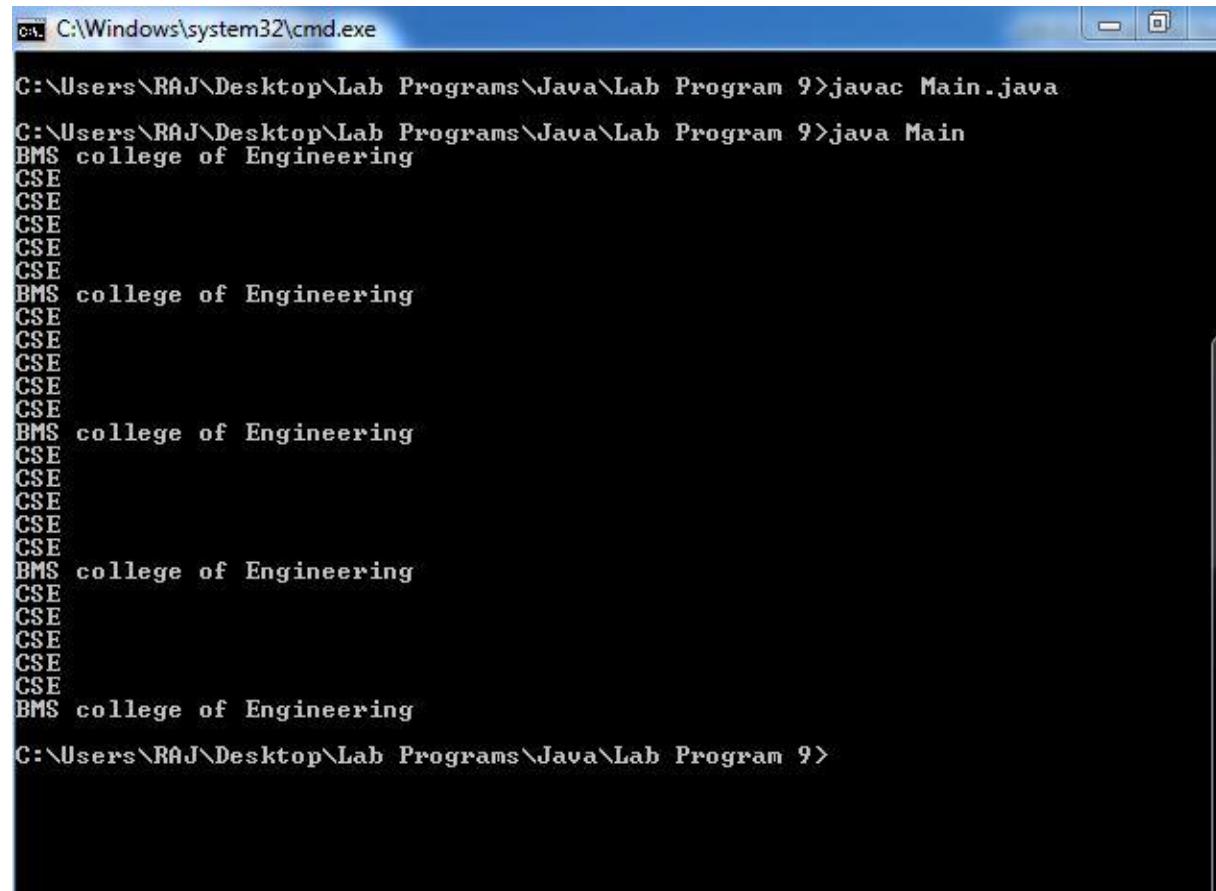
```
    Thread2(String name){
```

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

```
}
```

```
}
```



The screenshot shows a Windows Command Prompt window titled 'cmd C:\Windows\system32\cmd.exe'. The command entered is 'C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 9>javac Main.java'. The output of the compilation process is displayed below, showing multiple instances of the class 'Main' being compiled, each printing the text 'BMS college of Engineering' followed by 'CSE' five times. The final command entered is 'C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 9>'.

```
C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 9>javac Main.java
C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 9>java Main
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
C:\Users\RAJ\Desktop\Lab Programs\Java\Lab Program 9>
```

## Lab Prog 9

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("USE");

Thread.sleep(2000);

}

catch(InterruptedException e){

System.out.println(e);

}

3

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(1000);  
}
```

```
3 catch (InterruptedException e) {  
    System.out.println(e);  
}
```

3

3  
class Main {

```
public static void main (String [] args) {  
    Thread obj2 = new Thread2 ("College Name");  
    Thread obj1 = new Thread1 ("Dept. Name");
```

try {

obj1.t.join();

obj2.t.join();

```
} catch (Exception e) {
```

System.out.println ("InterruptedException");

3

3

3

#### 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 ArithmeticException. Display the exception in a message dialog box.

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

public class Main extends Frame implements ActionListener{
    TextField t1,t2;
    String msg="";
    Button btn;
    Main(){
        Label l1 = new Label("First Number: ",Label.RIGHT);
        t1 = new TextField(10);
        Label l2 = new Label("Second Number: ",Label.RIGHT);
        t2 = new TextField(10);
        btn = new Button("Submit");
        this.add(l1);
        this.add(t1);
        this.add(l2);
        this.add(t2);
        this.add(btn,BorderLayout.CENTER);
        this.setVisible(true);
        this.setSize(600, 300);
        this.setLayout(new FlowLayout(FlowLayout.CENTER,20,10));
        btn.addActionListener(this);
        addWindowListener(new MyWindow());
    }
}
```

```
        setBackground(Color.BLACK);

    }

@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 Main();

}

}

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());

    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);

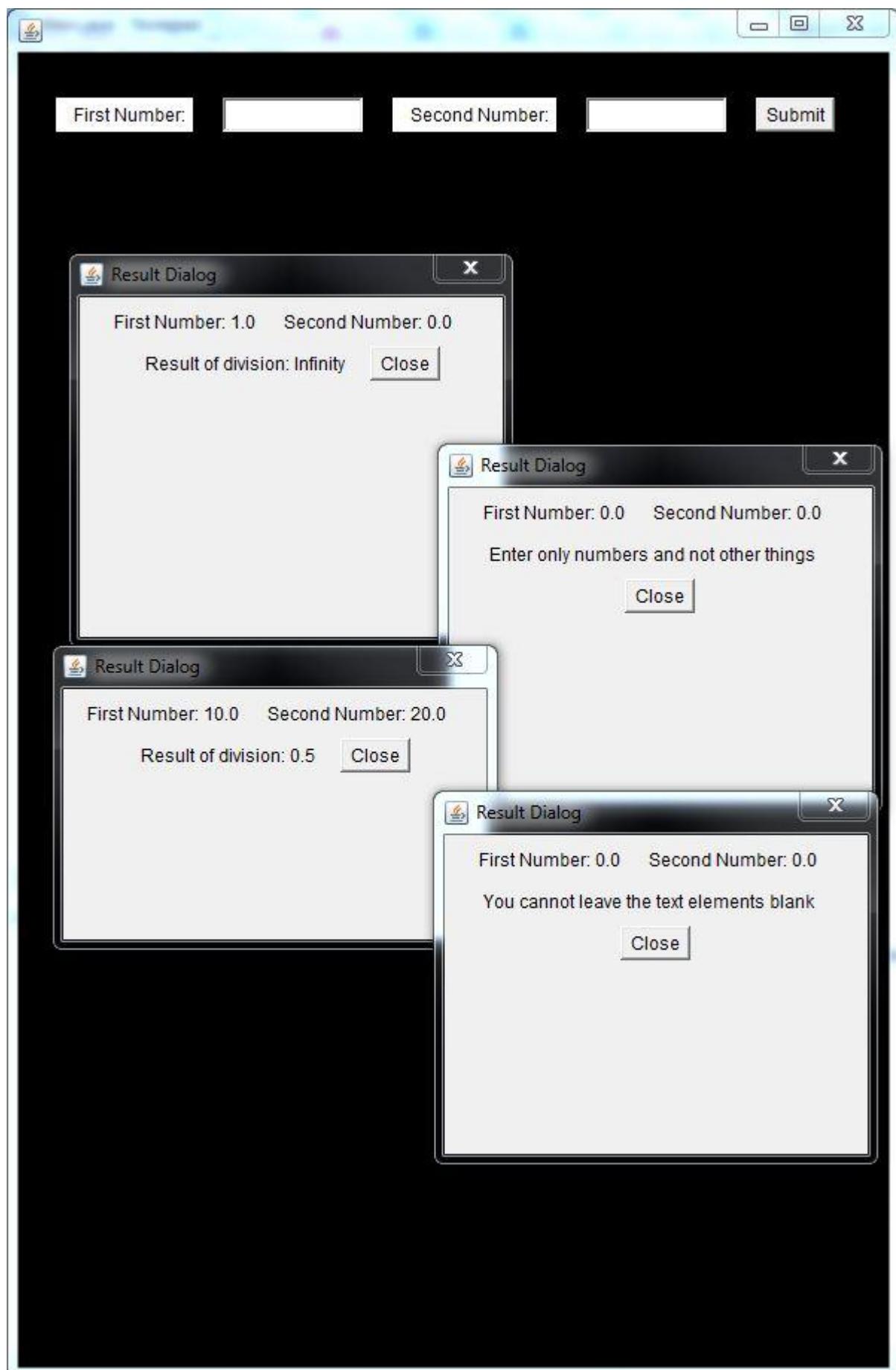
}
```

```
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);  
    }  
}
```



# LAB Prog 10

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

public class Main extends Frame implements ActionListener

Text Field t1, t2;

String msg = "4.

Button btn;

main()

```
Label l1 = new Label("First Number: 9",  
Label.RIGHT);
```

```
t1 = new TextField(10);
```

```
Label l2 = new Label("Second Number: 7",  
Label.RIGHT);
```

```
t2 = new TextField(10);
```

```
btn = new Button("Submit");
```

```
this.add(l1);
```

```
this.add(t1);
```

```
this.add(l2);
```

```
this.add(t2);
```

```
this.add(btn, BorderLayout.CENTER);
```

```
this.setVisible(true);
```

```
this.setSize(600, 300);
```

```
this.setLayout(new FlowLayout(CENTER,  
center,  
20, 10));
```

```
btn.addActionListener(this);
```

```
addWindowListener(new MyWindow());
```

```
setBackground(Color.BLACK);
```

3  
@ override

```
public Insets getInsets() {  
    return new Insets(50, 10, 10, 20);  
}
```

3

@Override

```
public void actionPerformed(ActionEvent e) {
```

```
String st1 = t1.getText();
```

```
String st2 = t2.getText();
```

```
n1 = 0.0;
```

```
n2 = 0.0;
```

```
if (st1.equals("a") || st2.equals("a")) {
```

msg = "You cannot leave the text elements  
blank".

3 else {

try 1

```
n1 = Double.parseDouble(st1);
```

```
n2 = Double.parseDouble(st2);
```

try 2

```
double res = n1 / n2;
```

msg = "Result of division: " + res;

```
3 catch (ArithmaticException e) {
```

```
msg = e.getMessage();
```

3

```
3 catch (NumberFormatException e) {
```

msg = "Enters only nos & not other  
characters";

3

```
3 new MyDialog(this, "Result dialog", false,
```

```
msg, n1, n2);
```

3

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

```
new Main(),
```

3

```
class mydialog extends Dialog implements ActionListener  
public mydialog (Frame owner, String title, boolean modal,  
String msg, double n1, double n2)  
    owner, title, modal);  
    this.setVisible (true);  
    this.setSize (300, 400);  
    this.setLayout (new FlowLayout());  
    this.add (new Label ("First Number: " + n1));  
    this.add (new Label ("Second Number: " + n2));  
    this.add (new Label (msg));  
    JButton b = new JButton ("Close");  
    this.add (b);  
    b.addActionListener (this);  
    this.addWindowListener (new WindowAdapter ()) {  
        public void windowClosing (WindowEvent e)  
            e.dispose();  
    };
```

@Override

```
public void actionPerformed (ActionEvent e) {  
    dispose();  
}
```

3  
class mywindow extends WindowAdapter {

```
public void windowClosing (WindowEvent e) {  
    System.exit(0);  
}
```

3