

SAI PRAVEEN MARNI
1BM19CS138

08/10/20

LAB-3

① Write a java program to find roots of quadratic equation

```
import java.util.Scanner;  
import static java.lang.Math.*;  
class quadratic  
{  
    public static void main(String args[])  
    {  
        quadratic obj = new quadratic();  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter the value of a ::");  
        float a = sc.nextFloat();  
        System.out.println("Enter the value of b ::");  
        float b = sc.nextFloat();  
        System.out.println("Enter the value of c ::");  
        float c = sc.nextFloat();  
        if(a == 0)  
    }
```

```
System.out.println ("Invalid");  
return;
```

}

```
float d = b*b - 4*a*c;
```

```
float sqrt_val = (float) Math.sqrt (abs (d));
```

```
float root1 = (-b + sqrt_val) / (2*a);
```

```
float root2 = (-b - sqrt_val) / (2*a);
```

```
if (d == 0)
```

{

```
System.out.println ("Roots are real and equal :: "+root1);
```

}

```
else if (d > 0)
```

{

```
System.out.println ("Roots are real and different \n");
```

```
System.out.println (root1 + "\n" + root2);
```

}

```
else
```

{

```
System.out.println ("Roots are complex\n");
```

```
System.out.print (-b / (2*a) + " + i" + sqrt_val / (2*a) +
```

```
"\n" + -b / (2*a) + " - i" + sqrt_val / (2*a));
```

}

}

* ALGORITHM :-

Step 1: START

Step 2: Input the value of a, b, c.

Step 3: calculate $d = b*b - 4*a*c$

Step 4: If ($d < 0$) Display "Roots are imaginary", calculate $r_1 = (-b + \sqrt{d}) / 2a$

and $r_2 = (-b - \sqrt{d}) / 2a$ else if ($d > 0$) Display "Roots

are equal" then calculate $r_1 = r_2 = (-b / 2a)$.

Step 5: print r_1 and r_2 .

Step 6: END.

→ Expected Output →

Enter the value of a :: 3

Enter the value of b :: 2

Enter the value of c :: 2

Roots are complex

- 0.333333334 + 10.745356

- 0.333333334 - 10.745356

09/10/20

LAB-4

SAIPRAVEEN MARIYAM
1BM19CS138

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 .

```
import java.util.Scanner;  
  
class student  
{  
    String USN;  
    String name;  
    int n;  
    double SGPA = 0;  
    int credits = 0;  
  
    Scanner in = new Scanner(System.in);  
  
    void Details()  
    {  
        System.out.println ("Enter the USN of the student");  
        USN = in.nextLine();  
        System.out.println ("Enter the name of the student");  
        name = in.nextLine();  
        System.out.println ("Enter number of subjects");  
        n = in.nextInt();  
        int credits = new int[n];  
        double marks = new double[n];  
        System.out.println ("Enter details of the subjects");  
        for (int i=0; i<n; i++)  
    }
```

```
System.out.println("Enter credits allotted to subject " + (j+1));
    credit[i] = in.nextInt();
System.out.println("Enter marks in subject " + (j+1));
    marks[i] = in.nextInt();
    calculate(credit[i], marks[i], i);
}
```

```
void calculate(int credit, double mark, int j)
```

```
{
```

```
    Credits = Credits + credit;
    if (mark >= 90 && mark <= 100)
        SGPA = SGPA + (10 * credit);
    else if (mark >= 80 && mark <= 89)
        SGPA = SGPA + (9 * credit);
    else if (mark >= 70 && mark <= 79)
        SGPA = SGPA + (8 * credit);
    else if (mark >= 60 && mark <= 69)
        SGPA = SGPA + (7 * credit);
    else if (mark >= 50 && mark <= 59)
        SGPA = SGPA + (6 * credit);
    else if (mark >= 40 && mark <= 49)
        SGPA = SGPA + (5 * credit);
    else
        System.out.println("Failed in subject " + (j+1));
```

```
} void Display()
```

```
{
```

```
    System.out.println("Details of student");
    System.out.println("Name : " + name);
```

```
    System.out.println("USN : " + USN);
```

```
    System.out.println("SGPA of student " + (SGPA / Credits));
```

```
    System.out.println("SGPA of student " + (SGPA / Credits));
```

```
}
```

public class sgpa

{
 public static void main (String args[])

{
 Student s1 = new Student();
 s1.Details();
 s1.Display();

}

}

* Output :-

Enter the USN of the student

1BM19CS138

Enter the Name of student

Praeern

Enter no of subjects

2

Enter the details of the subjects:

Enter credits allotted to subject 1

4

Enter marks in the subject 1

78

Enter credits allotted to subject 2

4

Enter marks in subject 2

87

Details of student

Name : Praeern

USN : 1BM19CS138

SGPA of student 8.5

"

16/10/20

LAB - 3

SAI PRAVEEN MARNI
IBM19CS138

Create a class Book which contains four members : name, author, price, num-pages. Include a constructor to set the values for the members. Include a toString() method that could display the complete details of book. Develop a JAVA program to create n book objects.

```
import java.util.*;  
  
class Book {  
    String name;  
    String author;  
    double price;  
    int num_pages;  
  
    Scanner in = new Scanner(System.in);  
  
    book() {  
        System.out.println("Enter name of the book");  
        name = in.nextLine();  
        System.out.println("Enter name of the author");  
        author = in.nextLine();  
        System.out.println("Enter price of the book");  
        price = in.nextDouble();  
        System.out.println("Enter no. of pages in book");  
        num_pages = in.nextInt();  
    }  
  
    public String toString() {  
        return ("Book name : " + name + " Author : " + author + " Price : " +  
               price + " Number of pages : " + num_pages);  
    }  
  
}  
  
class bookdetail {  
    public static void main(String[] args) {  
        int i, n;  
        Scanner in = new Scanner(System.in);  
    }  
}
```

```

System.out.println ("Enter no of objects to be created");
n = sc.nextInt();
book obj[];
obj = new book [n];
for (i=0; i<n; i++) {
    obj[i] = new Book();
}
System.out.println ("The details of the books are:");
for (i=0; i<n; i++) {
    System.out.println (obj[i].toString());
}
}

```

Output :

Enter the no of objects to be created:

2

Enter name of the book:

wolf hall

Enter name of the author:

william martel

Enter price of the book:

650

Enter no of pages in the book:

1500

Enter name of the book:

The corrections

Enter name of the author:

Jonathan Franzen

Enter price of the book:

450

Enter no of pages in book:

899

The details of the books are as follows:

Book name: Wolf Hall Author: Hilary Mantel Price: 650 No of pages = 1500

Book name: The Corrections Author: Jonathan Franzen Price: 450 No of pages = 879

6/11/20

LAB-4

SAIPRAVEEN MARAI
IBM19CS138

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 class extends the class shape. Each one of classes contains only the method printArea() that prints area of given shape.

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

```
abstract class shape
```

```
{
```

```
    int a=3, b=4;
```

```
    abstract public void print_area();
```

```
}
```

```
class rectangle extends shape
```

```
{
```

```
    public int area_rect;
```

```
    public void print_area()
```

```
{
```

```
    area_rect = a * b;
```

```
    System.out.println("The area of rectangle is " + area_rect);
```

```
}
```

```
}
```

```
class triangle extends shape
```

```
{
```

```
    int area_tri;
```

```
    public void print_area()
```

```
area-circle = (int)(3.14*a*a);
```

```
System.out.println("The area of circle is : " + area-circle);
```

```
}
```

```
}
```

```
class Circle extends Shape
```

REVIEW POINT

P-9A)

```
{
```

```
int area-circle;
```

better code for area of rectangle AVAL a good
better public void print-area()

{

System.out.println("The area of circle is : " + area-circle);

```
}
```

System.out.println("The area of circle is : " + area-circle);

```
}
```

System.out.println("The area of circle is : " + area-circle);

```
}
```

public class RTC

```
{
```

public static void main(String args[])

```
{
```

rectangle r1 = new rectangle();
r1.print-area();

triangle t1 = new triangle();
t1.print-area();

circle c1 = new circle();
c1.print-area();

```
}
```

(the output of algorithm for ans is "String. format")
Output :

The area of rectangle is : 12

The area of triangle is : 6 (the algorithm is)

The area of circle is : 28 (the algorithm is)

(the algorithm is)

06/11/20

LAB-5

SAIPRAVEEN MARNI
1BM19CS138

Develop a Java program to create a class Bank that maintains two kinds of accounts for its customers, one called savings account and 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 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-Accnt and Sav-Accnt to make them more specific to their requirements. Include the necessary methods in order to achieve following tasks:

- Accept deposit from customer and update the balance.
- Compute & deposit interest. Display the balance.
- Permit withdrawal and update the balance. Check for minimum balance, impose penalty if necessary and update the balance.

import java.util.Scanner;

```
class Account {  
    String name;  
    int accountNo;  
    String accountType;  
    double balance;  
    Account (String name, int accountNo, String accountType, double balance) {  
        this.name = name;  
        this.accountNo = accountNo;  
        this.accountType = accountType;  
        this.balance = balance;  
    }
```

this.name = name;

this.accountNo = accountNo;

this.accountType = accountType;

void DisplayStatus()

{

System.out.println("****" + this.accountType + "****");

System.out.println("Name : " + this.name);

System.out.println("Account No : " + this.accountNo);

System.out.println("Account Type : " + this.accountType);

System.out.println("Balance : ", + this.balance);

}

}

class SavAcct extends Account

{

double depositAmount;

double withdrawAmount;

SavAcct (String name, int accountNo, String accountType,

{

double balance)

super(name, accountNo, accountType, balance);

}

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

private void checkBalance()

{

if (balance < 0)

{

System.out.println("Transaction is not possible. Balance less than zero");

balance += withdrawAmount;

withdrawAmount = 0;

withdraw();

}

}

void CalInterest()

{

```
System.out.println("Interest to be added");
```

```
System.out.println("Annual rate of interest: 4.10");
```

```
System.out.println("Enter tenure in terms of year");
```

```
int tenure = input.nextInt();
```

```
balance = balance * math.pow(1.04, tenure);
```

}

```
void Deposit()
```

{

```
System.out.println("Enter the Deposit Amount");
```

```
depositAmount = input.nextDouble();
```

```
balance += depositAmount;
```

}

```
void Withdraw()
```

{

```
System.out.println("Enter the withdrawal amount");
```

```
withdrawAmount = input.nextDouble();
```

```
balance -= withdrawAmount;
```

```
checkBalance();
```

```
System.out.println("withdrawal Amount=" + withdrawAmount);
```

}

{

```
class CurrAcct extends Account
```

{

```
double minBalance = 1000;
```

```
double depositAmount;
```

```
double withdrawAmount;
```

```
static Scanner input = new Scanner(System.in);
```

```
CurrAcct(String name, int AccountNo, String accountType, double  
balance)
```

{

```
super(name, accountNo, accountType, balance)
```

}

```

private void checkBalance() {
    if (balance < minBalance)
        System.out.println("Transaction not possible. Balance less than minimum balance.");
    balance -= withdrawAmount;
    System.out.println("Do u still want to be transaction with added charges");
    String ans = input.next();
    if (ans.toLowerCase().equals("yes"))
    {
        balance -= (withdrawAmount + (0.05 * withdrawAmount) + 100);
        System.out.println("ALERT: Negative balance. In Service charge added : " + (0.05 * withdrawAmount));
    }
    else
    {
        withdrawAmount = 0;
    }
}

void Deposit()
{
    System.out.println("Enter Deposit amount");
    depositAmount = input.nextDouble();
    balance += depositAmount;
}

void withdraw()
{
    System.out.println("Enter the Withdrawal amount");
    withdrawAmount = input.nextDouble();
    balance -= withdrawAmount;
    checkBalance();
    System.out.println("Withdraw Amount = " + withdrawAmount);
}

```

```

public class BankTest
{
    public static void main (String args[])
    {
        Scanner in = new Scanner (System.in);
        System.out.println ("Enter the name");
        String name = in.next();
        System.out.println ("Enter Account No");
        int num = in.nextInt();
        int i=0;
        while (i < 2)
        {
            System.out.println ("Enter the account type\nIn curr - current acc."
                + "\nIn sav - savings acc. It And Balance.");
            String type = in.next();
            if (type.equals ("curr"))
            {
                double bal = in.nextInt();
                currAcct c1 = new currAcct (name, num, "Current Account", bal);
                c1.DisplayStatus();
                c1.Deposit();
                c1.DisplayStatus();
                c1.withdraw();
                c1.DisplayStatus();
            }
            else if (type.toLowerCase().equals ("sav"))
            {
                double bal = in.nextInt()
                SavAcct s1 = new SavAcct (name, num, "Savvy Account", bal);
                s1.DisplayStatus();
                s1.Deposit();
                s1.DisplayStatus();
            }
        }
    }
}

```

```
s1. CalInterest();  
s1. Displaystatus();  
}  
++;  
}  
in-close();  
}
```

Output :

Enter the name

praveen

Enter the account no.

1234567890

Enter the account Type

Curr - current acc.

Sav - Savings acct. And Balance .

curr

560000

** Current Account **

Name : praveen

Account no. : 1234567890

Account Type : current Account

Balance : 560000.0

Enter the Deposit Amount

12000

** Current Account **

Name : praveen

Account No : 1234567890

Account Type : Current Account

Balance : 57200000

Enter the withdrawal Amount

34000

withdraw amount = 34000.0

** Current Account **

Name: praveen

Account No: 1234567890

Account Type: Current Account

Balance: 538000.0

Enter the account type

curr - current acc.

SAN - savings acc. And Balance,

SAN

12345

** Savings Account **

Name: praveen

Account No: 1234567890

Account Type: Saving Account

Balance: 12345.0

Enter the Deposit Amount

123

** Savings Account **

Name: praveen

Account No: 1234567890

Account Type: Savings Account

Balance: 12468.0

Enter the withdrawal amount

3456

withdrawal Amount = 3456

** Savings Account **

Name: praveen

Account No: 1234567890

Account Type: Savings Account

Balance: 9012.0

Interest To Be Added

Annual rate of Interest: 4%

Enter tenure in terms of years: 3

** Savings Account **

Name : praveen

Account No : 1234567890

Account Type : Savings Account

Balance : 10137.274368

20/11/2020

LAB - 6

SAI PRAVEEN MARNI
IBM19CS138

PACKAGE LAB PROGRAM

"student.java"

```
Package cie;  
import java.util.*;  
Public class student  
{  
    Public String usn;  
    Public String name;  
    public int sem;  
    Public void read()  
{  
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter usn of student :");  
        usn = sc.next();  
        System.out.println ("Enter name of student :");  
        name = sc.nextLine();  
        System.out.print ("Enter semester :");  
        sem = sc.nextInt();  
    }  
}
```

"internals.java"

```
Package cie;  
import java.util.*;  
Public class internals extends student
```

```
{  
    Public int[] cie_m = new int[3];  
    Public void read()  
    {  
        super.read();  
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter CIE marks :");
```

```

        for (int i=0; i<3; i++)
    {
        System.out.println("Enter marks of the course "+(i+1)+":");
        cie_m[i] = sc.nextInt();
    }
}

public void display()
{
    System.out.println("USN of student is "+usn);
    System.out.println("Name of student is "+name);
    System.out.println("Semester of student is "+sem);
}

```

"externals.java"

package see;

import java.util.*;

import java.io;

import java.lang.*;

Public class external extends cie.student

{

Public int[] see_m = new int[3];

Public int[] marr;

Public void read()

{

Scanner sc = new Scanner (System.in);

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

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

{

System.out.print("Enter SEE marks of the course "+(i+1)+": ");

See_m[i] = sc.nextInt();

, , ,

"main_stu.java"

```
import java.util.*;
import java.io.*;
import java.lang.*;
import cie.*;
import see.*;

public class student_end
{
    public static void main(String[] args)
    {
        int n;
        Scanner sc = new Scanner(System.in);
        int final_mark;
        System.out.print("Enter number of students:");
        n = sc.nextInt();
        internal[] in = new internal[n];
        external[] ex = new external[n];
        internal ob1 = new internal();
        external ob2 = new external();
        ob2.mark = new int[n];
        for(int i=0; i<n; i++)
        {
            System.out.println("Enter details of student " + (i+1) + ":");
            in[i] = new internal();
            in[i].read();
            ex[i] = new external();
            ex[i].read();
        }
        System.out.println();
        for(int i=0; i<n; i++)
        {
            System.out.println("Details of student " + (i+1));
            System.out.println("USN of student is " + in[i].usn);
        }
    }
}
```

```

System.out.println("Name of student is " + in[i].name);
System.out.println("Semester of student is " + in[i].sem);
for(int j=0; j<3; j++)
{
    finalMark = in[i].cie - m[j] + ((ex[i].see - m[j])/2);
    System.out.println("Final mark of student " + (i+1) + " " +
        "in course " + (j+1) + " " + finalMark);
}
System.out.println();
}
}

```

OUTPUT :-

```

Enter Number of students : 2
Enter details of student 1 :
Enter USN of student : 123
Enter name of student : sai
Enter semester of student : 2
Enter the CIE marks :
Enter the marks of course 1 : 34
Enter the marks of course 2 : 36
Enter the marks of course 3 : 38
Enter the SEE marks :
Enter SEE marks of course 1 : 78
Enter SEE marks of course 2 : 80
Enter SEE marks of course 3 : 90
Enter details of student 2 :
Enter the USN of student : ram
Enter the name of student : 456
Enter the semester of student : 2

```

Enter the CIE marks:

Enter marks of course 1 : 34

Enter marks of course 2 : 39

Enter marks of course 3 : 40

Enter the SEE marks:

Enter marks of course 1 : 89

Enter marks of course 2 : 78

Enter marks of course 3 : 88

Details of student 1

USN of student is 123

Name of student is sai

Semester of student is 2

Final marks of student 1 in course 1 : 73

Final marks of student 1 in course 2 : 76

Final marks of student 1 in course 3 : 83

Details of student 2

USN of student is 456

Name of student is ram

Semester of student is 2

Final marks of student 2 in course 1 : 78

Final marks of student 2 in course 2 : 78

Final marks of student 2 in course 3 : 84

27/11/20

LAB - 7

SAIPRAVEEN MARI
IBM19CS138

Write a program to demonstrate generics with multiple object parameters.

```
import java.io.*;
import java.lang.*;
import java.util.*;

class gen<T>
{
    T ob;
    gen(T o)
    {
        ob = o;
    }
    T getob()
    {
        return ob;
    }
    void showtype()
    {
        System.out.println("Type of T is " + ob.getClass().getName());
    }
}

class generic
{
    public void main (String [] args)
    {
        String n;
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the Integer number to be displayed using the generic style");
        n = sc.next();
        gen<Integer> ob = new gen<Integer> (Integer.parseInt(n));
        ob.showtype();
    }
}
```

```
int val = ob1.getob();
System.out.println("Value is : " + val);
System.out.println();
```

System.out.println("Enter the string to Be Displayed Using generic style");

```
n=sc.nextInt();
gen<String> ob2 = new gen<String>(n);
```

```
ob2.showtype();
```

```
String x = ob2.getob();
```

```
System.out.println("Value : " + x);
```

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

System.out.println("Enter the Double Number to Be Displayed Using generic style");

```
n=sc.nextInt();
```

```
gen<Double> ob3 = new gen<Double>(Double.parseDouble(n));
```

```
ob3.showtype();
```

```
double ans = ob3.getob();
```

```
System.out.println("Value : " + ans);
```

```
}
```

OUTPUT :-

Enter the Integer Number to Be Displayed Using generic style

10

Type of T is java.lang.Integer

Value is : 10

Enter the Double Number to Be Displayed using generic style

3.14216

Type of T is java.lang.Double

Value is : 3.14216

Write a program that demonstrates handling exceptions in inheritance tree. Create a base class called "Father" and derived class "Son" which extends the base class.

In Father class, implement a construct which takes the age and throws the exceptional WrongAge() when the input age < 0. In son class, implement a constructor that cases both father and son's age throws an exception if son's age is \geq father's age.

```

import java.util.*;
class WrongAge extends Exception {
    int detail;
    WrongAge (int A) {
        detail = A;
    }
    public String toString () {
        return "Enter correct age "+detail+" is invalid";
    }
}
class Father {
    Public int age;
    Scanner in = new Scanner (System.in);
    Father () throws WrongAge {
        System.out.print ("Enter Father's age : ");
        age = in.nextInt ();
        if (age < 0)
            throws new WrongAge (age);
    }
}

```

class son extends father {

Scanner in = new Scanner (System.in);

int fage;

son (father f) throws WrongAge {

this.fage = f.age ;

System.out.print ("Enter son's age : ");

this.age = in.nextInt();

if (this.age < 0)

throw new WrongAge (age);

if (this.age > f.age)

throw new WrongAge (age);

}

}

class ages {

public static void main (String args[]) {

try {

father f = new father();

son s = new son(f);

}

catch (Exception e) {

System.out.println (e);

}

}

}

OUTPUT :-

Enter the father's age : 40

Enter the son's age : 56

enter correct age 56 is invalid.

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 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 LabProgram9 {
    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 :-

BMS college of Engineering

CSE dept

CSE dept

CSE dept

CSE dept

BMS college of Engineering

CSE dept

CSE dept

CSE dept

CSE dept

BMS college of Engineering

CSE dept

CSE dept

CSE dept

CSE dept

BMS college of Engineering

CSE dept

CSE dept

CSE dept

CSE dept

BMS college of Engineering

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in 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 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;
    Label l1;
```

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");
//Label l = new Label("Updates:");
l1.setBackground(Color.YELLOW);
l2.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 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 myWindows());
setBackground(Color.YELLOW);
//System.out.println(BorderLayout.CENTER);
}

```

Y

@Override

```
public Insets getInsets() {
```

```
    return new Insets(50, 10, 10, 20);
```

Y

@ OnClick

```
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 (ArithmaticException 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 labo();
}
```

```

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 l1 = new Label("Updates on the result : ");
        //l1.setSize(300, 20);
        this.add(l1);
        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);
    }
}

```

Output :

Result Dialog

Updates on the result :

First Number : 450.0 Second Number : 30.0

Result of division : 15.0 [Close]